Jonna Halmela

Improving Test Management for Service Pack Release

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Industrial Management

Bachelor’s Thesis

16 November 2018

Abstract

Authors Title Number of pages Date	Jonna Halmela Improving Test Management for Service Pack Release 52 pages + 5 appendices 16 November 2018
Type of Project	Bachelor of Engineering
Degree Programme	Industrial Management
Professional Major	International ICT Business
Instructor	Anna Sperryn, Senior Lecturer

This Thesis focuses on improving test management in one team. The case company is a large manufacturing company which has a wide IT organization. The case team develops solutions to installers and their supervisors. Testing those solutions was unorganized and therefore harmonizing testing between the team, company and vendors was needed.

The study is conducted as a case study where data is collected from interviews, workshops and internal documents. Topics for literature study were selected based on the weaknesses identified from the current state analysis. The proposal focuses on solving the weaknesses using the best practices and professional insight.

The proposal suggests the ways to improve the test management in the case team. By deploying a test process, establishing roles, documentation and test cases, the testing becomes more efficient and it has better quality. The improved test management can also help to reduce incidents and increase customer satisfaction. The study can be used also on the company level to help other teams to organize their testing management.

Keywords	ITIL, Test Management, Testing, Test Case, Process Implementation, Test Management Tool

Tekijä Otsikko Sivumäärä Aika	Jonna Halmela Sovelluspäivityksen testaushallinnan parantaminen 52 sivua + 5 liitettä 16.11.2018
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tuotantotalous
Suuntautumisvaihtoehto	Kansainvälinen ICT-liiketoiminta
Ohjaaja	Lehtori Anna Sperryn

Tämä insinöörityö keskittyy parantamaan yhden tiimin sovellustestauksen hallintaa. Kohdeyritys on suuri suomalainen teollisuuden toimija, jolla on laaja IT-organisaatio. Kohdetiimi kehittää ratkaisuja asentajille ja heidän valvojilleen. Näiden sovellusten testaaminen on ollut organisoimatonta, joten testauksen yhtenäistäminen tiimin, yrityksen ja toimittajien kesken on tarpeellista.

Insinöörityö on tehty tapaustutkimuksena, jossa tutkimusmateriaali on kerätty haastatteluista, työpajoista ja sisäisistä dokumenteista. Nykytila-analyysi tutkii tiimin nykytilaa ja tiivistää heikkoudet sekä vahvuudet. Nämä ovat pohjana kirjallisuustutkimuksen aiheille, jotka ovat ITIL, testaaminen ja testaushallinta. Kehitysehdotus keskittyy ratkaisemaan heikkoudet käyttämällä parhaita käytänteitä ja haastattelusta saatuja näkemyksiä.

Kehitysehdotus esittelee tapoja parantaa sovellustestaushallintaa tiimissä. Kehittämällä testausprosessia, selkeyttämällä rooleja, lisäämällä dokumentaatiota ja ottamalla käyttöön valmiit testitapaukset testauksesta tulee tehokkaampaa ja laadukkaampaa. Testaushallinta voi myös vähentää virheiden määrää sovelluksissa ja nostaa asiakastyytyväisyyttä. Insinöörityötä voidaan käyttää myös yritystasolla auttamaan muita tiimejä organisoimaan testauksensa.

Avainsanat	ITIL, Testaushallinta, Testaus, Testitapaukset, Prosessi-implementaatio, Testaustyökalu

Contents

(Due the confidentiality of the materials, Appendices are only for the company’s use.)

List of Key Concepts

ITIL	Information Technology Infrastructure Library is a collection of frameworks and best practices to manage IT services
Test management	Managing the whole software test process
UAT testing	User Acceptance Testing, where users and owner test the usability of the solution
Regression testing	Testing which confirms that all component works together
Incident ticket	Service desk ticket that implicates that something (application, login, feature) does not work
Service pack	In this thesis refers to update that brings new features to the solution
IRMA	Installation Resource Management Tool, browser-based solution for supervisors
IBR	Installation Back reporting Tool, mobile solution for installers and supervisors
IMT	Installation Management Tool, a mobile solution for installation supervisors
IES	Installation Execution System is a mobile solution for installers and installation subcontractors
IOS	Installation Operation Solutions, the case team

1 Introduction

Test management makes an essential part of the software development and the key to the project success. It ensures that a solution fulfills requirements and has high quality. Testing verifies that performance, user experience, and security are on the required level. This decreases incidents and customer complains. Proper testing will also reduce costs in all parts of development.

This study focuses for improving test management in one team. Their main challenge was an unstructured process and undefined roles. Business context, challenge, outline, basic concepts, and structure of the study are introduced below.

1.1 Business Context

The case company of this thesis project is a Finnish industrial enterprise which is globally one of the largest in its area. The annual net sales are 8.9 billion Euros (2017) and the company delivers over 25,000 new solutions every year worldwide. The case company operates in 50 countries, focusing on new solutions, maintenance, and modernization. It has 55,000 employees around the world. This includes, for example, manufacturing, installation, administration, and design.

IT department of the company has hundreds of employees and several teams. During the last four years, IT department has created IT Operating Model based on ITIL framework that covers all units, processes and roles. There is also an ongoing process to implement the model to teams. The department has streamlined structures and speed solution development.

This study focuses on Installation Operation Solution (IOS) team, which is one of the four teams in Delivery solutions. The case team manages solutions for supervisors and installers in the field. At the moment, they have three applications and one in progress. Altogether, the team has 10 experts in Finland, India, Italy and the United States.

1.2 Business Challenge

The business challenge relates to the **lack of structured test management process** in the IOS team. Firstly, unclarified roles have led to the rollout coordinators using excessively time for testing, since managing testing was no one's responsibility. Absence of a test manager resulted in the lack of test cases, reporting and planning. Second reason for disorganized testing was the use of several vendors and project platforms. This has caused dispersed documentation and various ways of testing. Also, the team was not sure how comprehensive testing their vendors do. Thirdly, **the case team has not implemented the case company's IT Operating model** which guides how to arrange testing.

As a result, defective testing management affects the employees' performance and perception of meaningfulness of their job. They feel that testing takes time from the actual job. Therefore, rollouts for the new areas slowed down and customer did not get the support as much as they needed. Also, solution development decelerated when a vendor waited for testing results, or new versions of applications were released incomplete.

1.3 Objective, Outcome and Scope

The objective of this study is to **harmonize testing processes together with the team, vendors, and the company.** The outcome is a start of implementation of the IT Operating Model's test management process in the case team, and the test cases for all applications in the selected test management tool. The proposed test process includes the defined roles, a release calendar, and the vendor's responsibilities.

In this thesis project, improving test management was limited to the case team and its stakeholders. The project does not include other units, or changing the case company's processes. The study includes a closer look at the IOS team's and vendors' actions, a start of the implementation of the proposed test management process, and some proposed practical improvements such as test cases and a test management tool.

1.4 Thesis Outline

The study is conducted in three parts. The first part included interviews, workshops, and getting familiar with the internal and external documentation and portals. This created the picture of the current state of test management in the case team. In the second part, there were interviews and studying the case company's material which led to the proposal for improving test management. The last part included collecting feedback for the validation.

This Thesis is written in 7 sections. Section 1 is the Introduction. Section 2 overviews used methods and data gathering. Section 3 analyses the current state of test management. Section 4 discusses the best practices of testing. Section 5 presents proposal. Section 6 briefly tells about the results of validation. Finally, Section 7 includes the summary, next steps and the evaluation of the Thesis project.

2 Method and Material

This section describes the methods and materials of the study. Design, project plan and research design open how the study is conducted.

2.1 Research Design

This study was conducted in 5 stages shown in the research design diagram below.

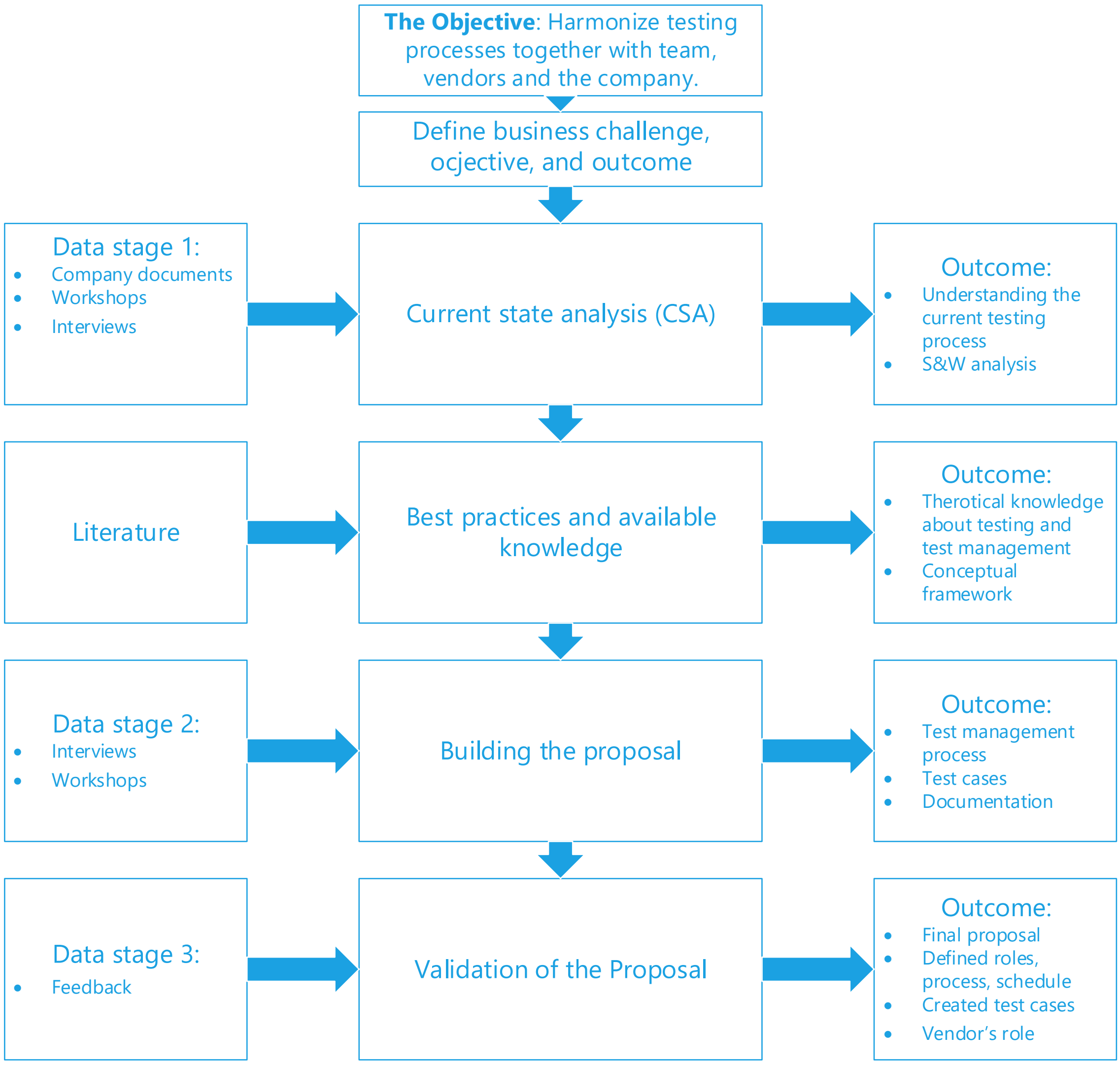

The Objective: Harmonize testing processes together with team, vendors and the company.
Define business challenge, ocjective, and outcome
Data stage 1:
Company documents
Workshops
Interviews
Current state analysis (CSA)
Outcome:
Understanding the current testing process
S&W analysis
Literature
Best practices and available knowledge
Outcome:
Therotical knowledge about testing and test management
Conceptual framework
Data stage 2:
Interviews
Workshops
Building the proposal
Outcome:
Test management process
Test cases
Documentation
Data stage 3:
Feedback
Validation of the Proposal
Outcome:
Final proposal
Defined roles, process, schedule
Created test cases
Vendor's role

Figure 1. Research design for Improving test management

As seen in Figure 1, defining the business challenge, objective and outcome was the first step and the basis of the study. Data 1 was used for gathering the understanding of the current state. The outcome of this step was the summary of strength and weaknesses analysis which helped understand the current testing process and identified the topics for improvement and for exploring the theory.

In the next stage, the study investigates the best practice and available knowledge on testing and test management. The goal for this stage was to identify applicable knowledge and best practice, and later apply it to solving the business challenge. The outcome of this stage is the conceptual framework, a construct that shows the key elements of best practice and knowledge needed for building the solution.

Based on the findings from the current state analysis, combined with best practice identified from literature, this study builds a proposal for improving the test management process. The proposal includes four areas: Test management process and roles, documentation, test cases for regression testing and suggestions to IES testing.

Finally, after the proposal was built, it was demonstrated to the IOS team and discussed during the final presentation session. Results from the final presentation session were included as the company feedback when building the final version of the proposal.

2.2 Project Plan

The research part of the study was conducted during summer 2018 together with on the case team, and the thesis report was finished later independently in autumn 2018. The schedule for the summer can be seen in Figure 2 below.

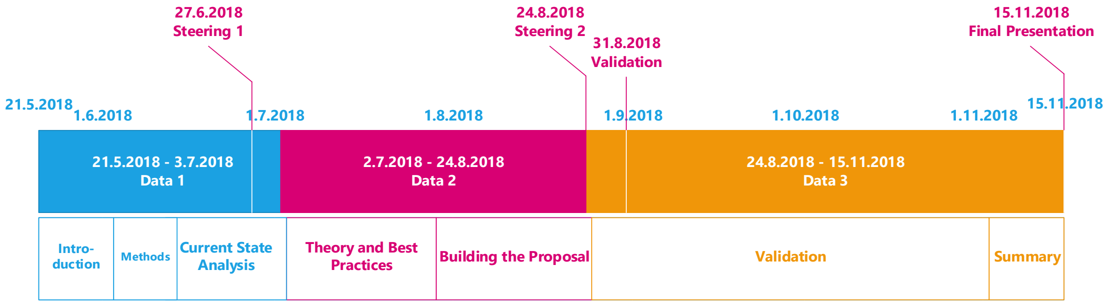


Figure 2. The project schedule

As seen in Figure 2, during the first six weeks of the research part of the project Data 1 was collected, and used for conducting the current state analysis. Data 2 was gathered in the next eight weeks for building the proposal. Last months were used for collecting feedback (Data 3) for validation and finalizing the proposal.

Thus, this project makes a real life business project, conducted in the company setting, and based a variety of information inputs and analysis of multiple data sources. The data sources and data analysis are described in detail below.

2.3 Data Collection and Analysis

Data of the thesis project was collected in three rounds and from a multiply data sources. Data is shown in Table 1 below.

Table 1. Details of interviews, workshops and discussions, in Data 1-3.

	Participants	**Data type**	**Topic, description**	**Date, length**	**Documented as**
	Data 1, for the Current state analysis (Section 3 or 4)				
1	Delivery solution	Workshop	Annual meeting of delivery solutions.	29.-31.5.2018	
2	Team	Workshop	Kick-off workshop for implementing IT operational model	7.6.2018, 1,5 hours	Field notes
3	Team	Workshop	Requirement and Release management	15.6.2018, 1,5 hours	Field notes
4	Team	Workshop	Change, Test and Configuration management	19.6.2018, 1,5 hours	Field notes
5	Team	Workshop	Summary with internal expert	26.6.2018, 1,5 hours	Field notes
6	Solution owners (3 persons)	Skype Interview & Face to-face interview	Benchmarking with other Delivery solution teams' solution owners.	June 2018, 4 hours	Field notes and recording
	Data 2, for Proposal building (Section 5)				
7	Internal expert 1	Face to face Interview	Test process and test management in case company	5.6.2018, 1 hour	Field notes
8	Internal expert 2	Face-to-face Interview	Test process and test management in case company	13.6.2018, 0,5 hours	Field notes
9	Internal expert 1	Face to face Interview	Test management tool	25.6.2018, 1 hours	Field notes

10	Team	Workshop	Internal expert + follow up	10.8.2018, 1,5 hours	Field notes
11	Team	Workshop	Release calendar	14.8.2018, 1,5 hours	Field notes
	Data 3, from Validation (Section 6)				
12	Team	Workshop	Test management tool demo and discussion	24.8.2018 0,5 hours	Field notes
13	Team	Workshop	Validation, evaluation of the Proposal	31.8.2018, 1 hour	Field notes
14	Team	Workshop	Test case validation and datasets	5.9.2018, 1 hour	Field notes

As seen from Table 1, data for this project was collected in three rounds. The first round, for collecting Data 1, was conducted for the current state analysis. Workshops were discussions between the team to understand the current situation, and especially what should be done to implement the IT operating model to the test, release, change and requirement management.

In the next round, Data 2 was collected to gather suggestions from the case company and the team for developing the proposal. This data included interviews and workshops. The final data was collected when receiving feedback for the proposal from the case team.

In this study, the interviews and workshops made the primary method of data collection. The interviews were conducted as face-to-face interviews, held on the company premises, with questions created in advice. The workshops were part of the case team's IT operating model implementation process.

Besides interviews and workshops documentation, guides and incident tickets were data sources. They are listed in Table 2Table 1.

Table 2. Internal documents used in the current state analysis, Data 1.

	Name of the document	Number of pages	Description
A	The case company’s IT Operating Model, v2.0.pdf	69 pages	A high-level summary of the case company’s IT Operations
B	The case company testing handbook	15 pages	Diagrams for Operational Processes, ppt.
C	SharePoint site	6 folders	Team’s documentation in team site
D	IMT Guide	51 pages	Application handbook, ppt.
E	IRMA Guide	54 pages	Application handbook, ppt.
F	IBR Guide (installers, supervisors)	21 + 21 pages	Application handbooks, ppt.
G	Incident tickets	492 tickets	Incident tickets from Remedyforce

As seen from Table 2, this project also analyzed several internal documents. The main documents included team documentation, the IT Operational Model and Incident tickets. The documents were analyzed for understanding the current situation of testing management and documentation. The documentation was also used for building the proposal. Most of the data analysis was conducted in the current state analysis stage.

IMT, IRMA and IBR Guides were used for learning how to use the applications and creating test cases. The findings from the current state analysis are discussed in Section 3 below.

3 Current State Analysis

This section discusses the results from the current state analysis of the test management process. The background and the map of the current process help to understand the case team. Findings from the current process is based on Data 1.

3.1 Overview of the CSA Stage

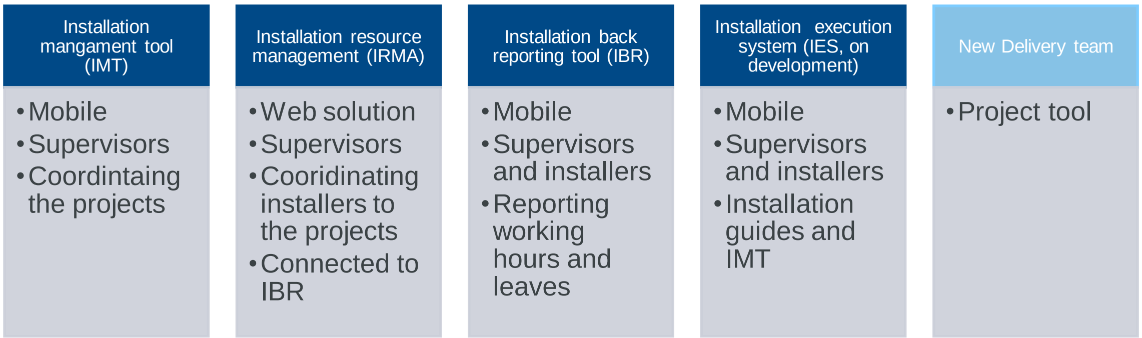
The current state analysis was conducted in three steps. Firstly, collecting information from internal documents, vendors' project portals, and the solutions. This gave insight how applications work and what they include. Also, reviewing documentation increased apprehension of the testing practices.

The second area of data collection included workshops and the key stakeholders' interviews. In the workshops, the team gathered together to share thoughts about IT Operating Model's test, release, change, and requirements management. These four sessions gained valuable knowledge to all participants and data to the study. The interviews with the key stakeholders clarified the current testing process. Last step was to analyze all gathered information and formulate the findings from the current state analysis.

3.2 Background of Test Management Process

The case team was originally built around IMT (Installation management tool) application. IRMA (Installation resources management tool), IBR (Installation back reporting tool), IES (Installation execution systems), and Project tool are developed over the years for the installers' needs. Employees have also changed during different developing phases. The project tool was separated into own team in January 2018. These practices, as well as a utilization of multiple tools, have led to the roles and processes being undefined.

The above mentioned applications are made for mechanical installers and their supervisors. Besides end users, applications have key-users on frontlines. They communicate with the IOS team and are responsible for usage. Figure 3 below shows the structure of the applications.



Installation mangament tool (IMT)	Installation resource management (IRMA)	Installation back reporting tool (IBR)	Installation execution system (IES, on development)	New Delivery team
• Mobile • Supervisors • Coordintaing the projects	• Web solution • Supervisors • Cooridinating installers to the projects • Connected to IBR	• Mobile • Supervisors and installers • Reporting working hours and leaves	• Mobile • Supervisors and installers • Installation guides and IMT	• Project tool

Figure 3. Applications of the team

As seen from Figure 3, the first column, IMT, includes a mobile application and the features variates by a country. It has two vendors, one for SAP integration and other for the user interface. IMT3 is the same application than IMT2 but built in different platform to ensure more stable and reliable user experience. Old features are moved to IMT3 one-by-one and therefore there is no actual requirement-based development going on. Users have access to both versions.

Next, IRMA includes a web application which has a few service pack releases yearly. As shown in the third column, IBR includes a mobile application, with own views for installers and supervisors. Development is currently in the pilot state. IRMA and IBR has the same vendor who is responsible for the whole package. Finally, the newest mobile application is IES which development has not started yet.

The IOS team's software development is based on the continuous development idea. This means to launch small and do small enhancements often. This way development will not take years and it gains better understanding to what end-users need. It is part of IT Operating Model's Development area. In this thesis, all updates that adds new features are called service packs.

The current state analysis focused on Service transition which carries out the development to production. IT Operating Model hierarchy is seen in Figure 4.

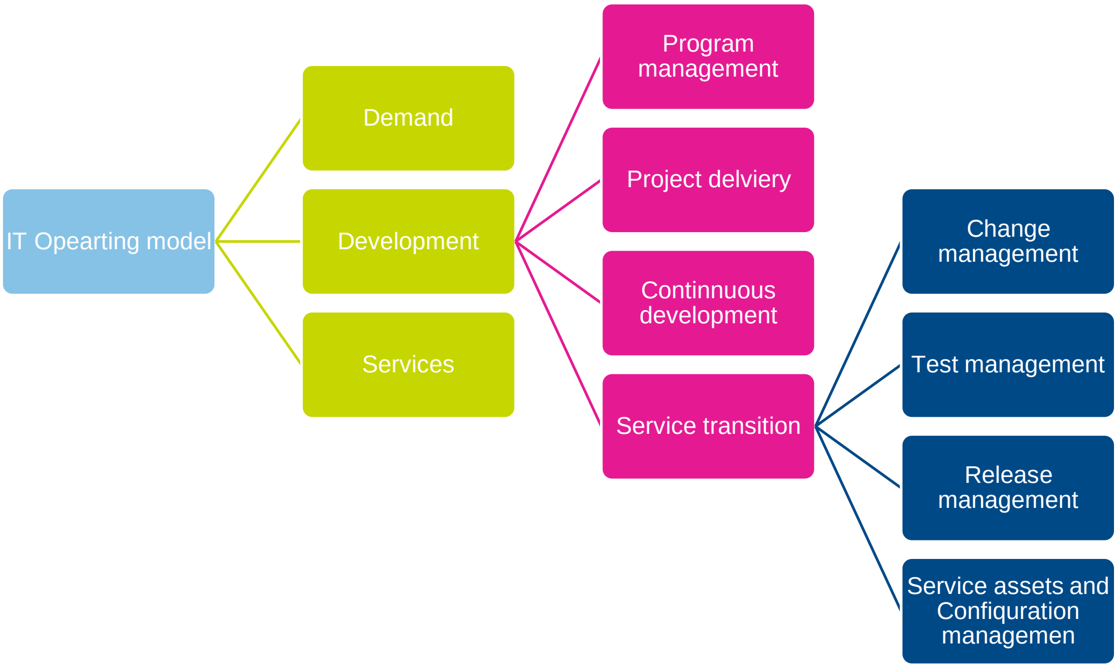


Figure 4. Structure of IT Operating Model

IT Operating Model is the guideline to whole IT department in the case company. It is based on ITIL's (Information Technology Infrastructure Library) best practices. As seen from Figure 4, the model has three areas: Demand, Development and Services. Development includes Program management, Project delivery, Continues development and Service transition. Change management, Test management, Release management and Service assets and configuration management are in Service transition. Every team carries out the process implementation in own schedule and practice.

Based on the interviews, there is a plan to centralize the testing unit for some solutions starting from 2019 but the IOS team is not part of the trial at this point. The case team will implement requirement, release and change management during Autumn/Winter 2018.

3.3 Map of the Current Service Pack Release Process

Due to the use of different vendors, release processes are different for IMT and IRMA (and IBR). IRMA has more structured testing which is shown below in Figure 5.

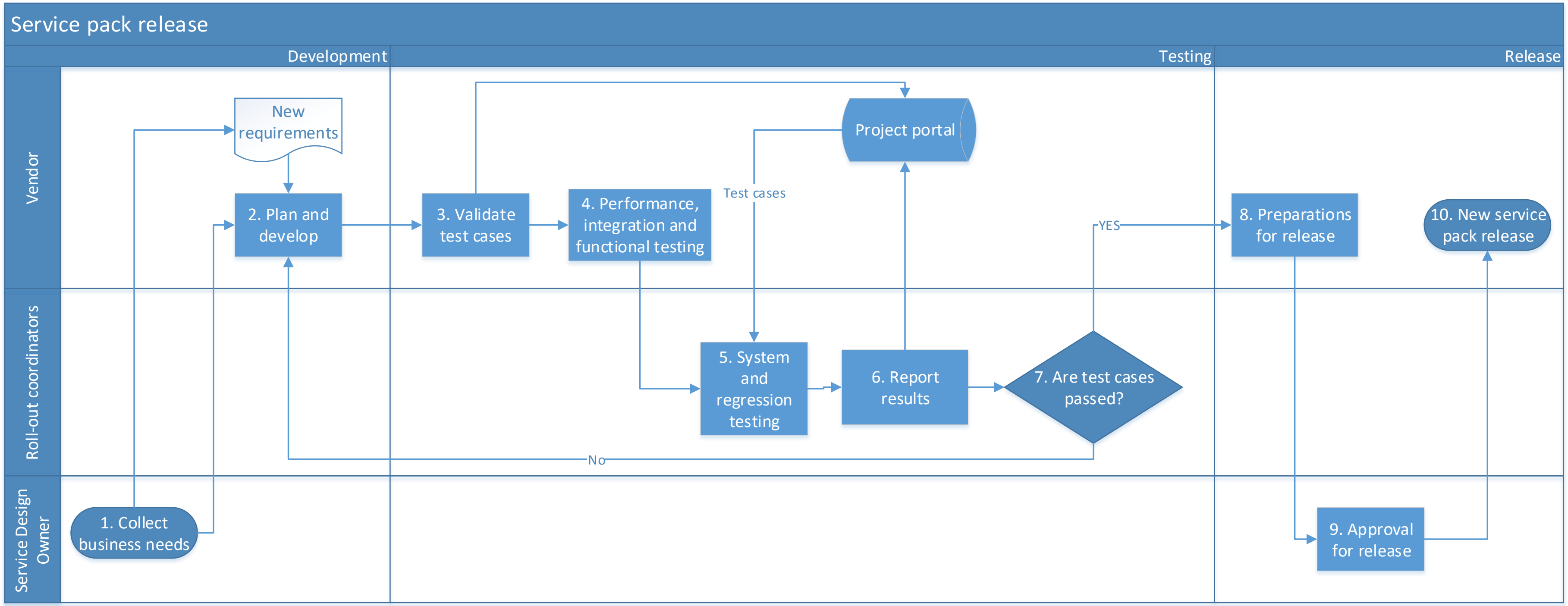


Figure 5. Service pack release process for IRMA

As seen in Figure 5, the new service pack process starts when Solution design owner collects business needs for requirements (box 1). They make the basis for development and later testing. The vendor plan and develop changes (box 2) to the solution are based on the requirements and faults in the previous version. Next step is the validation of the test cases (box 3). The validation covers all changes and new functionalities. Test cases are stored in the vendor's project portal and sent to the rollout coordinators by email.

The vendor is responsible for performance, integration, and part of the system testing (box 4). After those steps, the vendor gives a direction to the case team's rollout coordinators what to test and how. Coordinators have access to the project portal to see and execute the test cases (box 5) and notify errors (box 6). Main responsibility of the rollout coordinators is the functional and regression testing. Developer corrects issues and the coordinators test the application again. This happens usually in three sprints (box 7). The vendor does final polishing and preparation for the release (box 8). After the solution design owner has given his approval for the launch (box 9), it is possible to release a new service pack for end-users (box 10). After the service back is released, the process starts again. Rollout coordinators may collect feedback to review the changes and understand what the front-lines want.

IMT testing is not as managed as IRMA testing. Instead of the prepared test cases, the vendor gives a list of features that should be tested. Team members test IMT randomly and report errors into the vendor's project portal.

3.4 Findings from the Current State of Test management Process

Discussion with the team and closer look at the documentation and incident tickets made the basis of the following findings.

3.4.1 Documentation

The case company and team use SharePoint for sharing documentation in-house and with key users. All applications have own folders on the team's site where newsletters, guides and support information are placed. The team's site has also an admin folder for internal use. It includes, for example, comprehensive handbooks, development materials and technical information. Folders, however, are not updated regularly.

A check of the requirement documentation has shown that it is too defective for testing. These documents are not listed in one place or numbered, therefore it is not possible to do regression testing based on these requirements. Also, all features are not defined clearly on paper. This have led to lazy solutions for development, especially for IBR which is unfinished. In future, the requirement collection will happen on Remedyforce, which is the IT portal of the case company. There, key-users can suggest new features and give feedback.

Both vendors have their own project portals where are the documentation, meeting notes and test cases are stored. There, the rollout coordinators can also report errors and follow the fixing process. Vendors do not provide test reports after releases; therefore the case team feels uncertain about comprehensiveness of testing. Vendor's developers have also changed frequently, hence their knowledge of solution may be too restricted for testing and understanding all pitfalls. They do not send anymore screenshots and other material for user guides which increase the team member's workload.

3.4.2 Structure and roles

The case team does not have a specific test manager or testers, and testing is not part of anyone's job description. Therefore, two rollout coordinators conduct IRMA and IBR testing and IMT testing is for three employees in service area. This includes part of the system testing, regression testing and UAT testing before every service pack release. Key-users can be part of the UAT testing. Sometimes employees do the testing between development sprints or before and after SAP releases. The scope is to ensure that application and new features function as agreed.

Developers give list of features to test and testers report errors back. Problem is that IMT does not have test cases and IRMA's test cases cover only the new features. Due to this, the testers need to know how to use the application and how it should work in special cases. Regression testing is not structured, therefore not all basic features are tested with every service pack release.

Also, the case team do not have explicit release and test calendar for medium releases and service packs. Consequently, employees do not have enough time to test and report errors before scheduled release. Lack of time and the unstructured testing cause incidents, customer discontent and need for more development. Based on interviews and workshops, the employees are eager to improve and have recognize the need for change. However, team members do not have time or knowledge to tackle the problems.

3.4.3 IMT testing

IMT has own set-ups and languages for each region and country. This indicates that one feature can work in Finland but not in Italy. Therefore, testing needs to be carried out with different dataset. This enquires time and occasionally errors are not noticed during testing. Frontlines and key-users do the UAT testing as part of the roll-out. The idea is to train key-users and find last faults before the local release.

IMT has two vendors and the user-interface vendor does not have access to SAP. The IOS team does all testing and communication between vendors. Therefore, IMT2 and IMT3 cause more incidents than IRMA. Login and application issues cause most of the incidents, as seen in Figures 6 and 7 below.

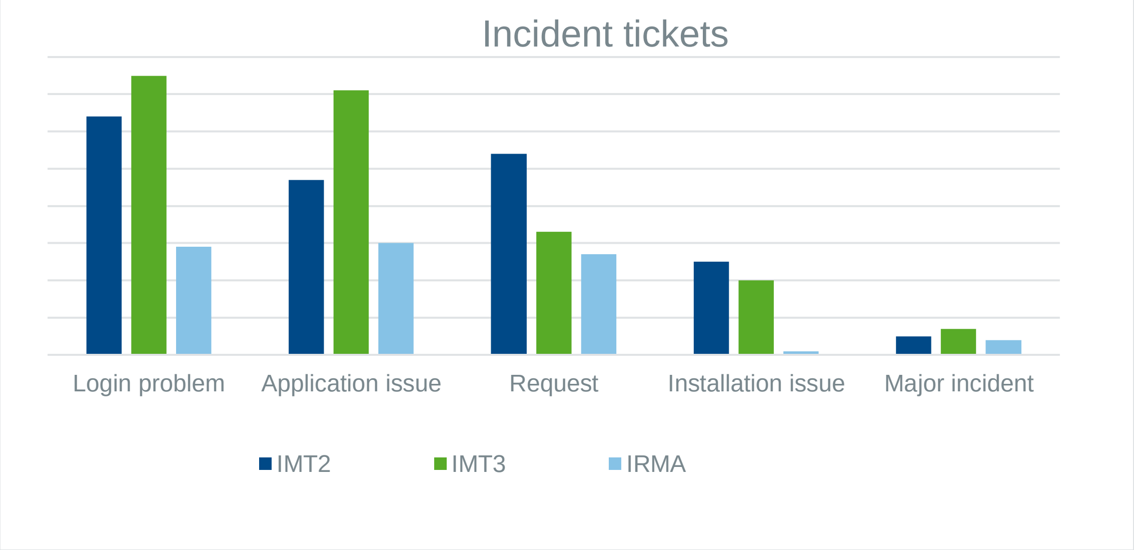


Figure 6.   IMT and IRMA Incident tickets (1.1.-17.7.2018)

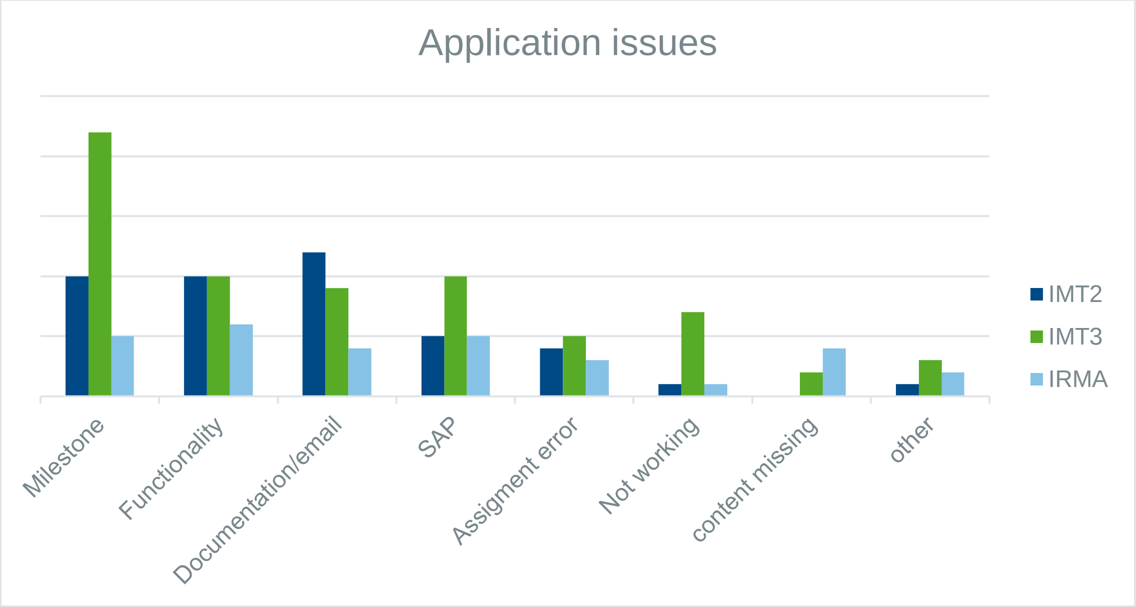


Figure 7. IMT and IRMA Application issues (1.1.-17.7.2018)

For the analysis of the situation, incident tickets were divided into 5 sections (as seen in Figure 6). Login problems and installation issues are mainly result from inoperative of SAP. A request should not be an incident tickets rather request ticket. Major incidents are incident that affect to several users, usually to the whole country. These incidents were primarily login problems or relate documentation/email.

The analysis also revealed that the application issues are the most relevant regarding to the testing. Milestones, functionality and documentation/email cause incidents (as seen in Figure 7). Part of those could have been avoided with a comprehensive testing.

3.5 Summary of Key Findings from the Current State Analysis

Current state analysis revealed the findings that pointed clearly to the strengths and weaknesses in the current testing practices. The results are summarized in Table 3 below.

Table 3. Strengths and weaknesses of testing management

Strengths	**Weaknesses**
SharePoint is used	Cooperation with vendors
Employees are ready to improve	Unstructured testing
IT operating model is launched at the company	IMT testing process is not thought
Employees are used to do testing	No clear roles
IRMA and IBR have test cases	Constricted test cases

As seen in Table 3, the strength of the current Test management is using SharePoint for documents. Also, the case company has already thought how to arrange the test management process, and the IT operating model is being launched across the case company. Importantly, the employees want to improve their performance with testing, and IRMA and IBR testing are already going to the right direction and developed test cases. Next step is to implement the test management process of the IT operating model to structure the testing.

Test management process, however, also have weaknesses. First of all, cooperation with vendors is not sufficient, there are also unclear roles internally in the team, and the limited test cases are all implication of unstructured IMT testing.

This Thesis focuses on improving test management process by creating the test cases and testing the schedule, putting documentation in order, and defining the roles. By tackling these weaknesses it will be possible to improve testing management in the case team.

Next, this study discusses the industry best practice and available knowledge for improving test management process.

4 Available Knowledge and Best Practice on Test Management

This section discusses the best practice to organize test management. Understanding basic concepts and best practice of testing and test management area will help create the theoretical framework for building the improvement proposal.

4.1 Test Management in ITIL

ITIL (Information Technology Infrastructure Library) is a collection of best practices for managing IT services through its lifecycle. Companies can implement the whole package and apply for a certificate. Other way is to modify ITIL to fit the company and use it as a tool box. ITIL includes five areas: Service Strategy, Service Design, Service Transition, Service Operation and Continual Service Improvement as seen in Figure 8 below (ITIL, 2007. p.3-7).

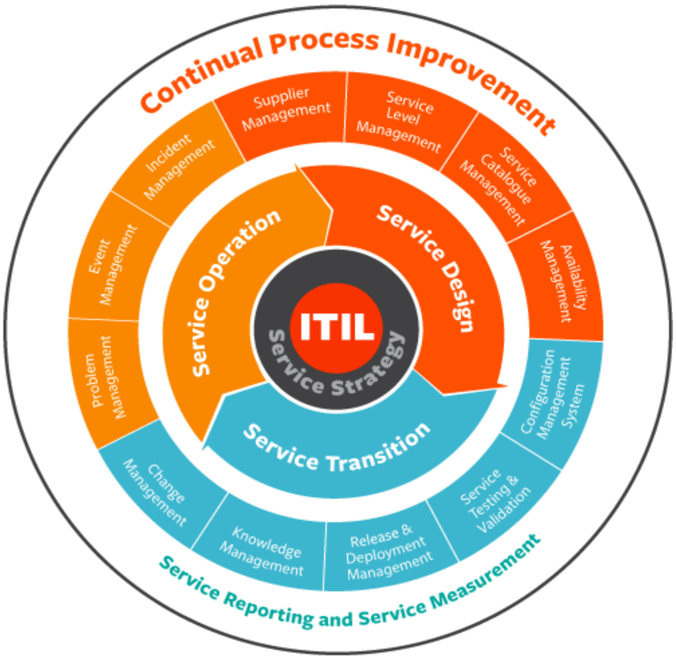


Figure 8. ITIL lifecycle (BMC, 2018)

Figure 8 shows the ITIL framework. Service strategy is the core of the functions, it designs, develop and implement Service management. Service design starts the process by managing suppliers, service levels, service catalogue and availability. Next phase is Service transition where the changes to the product or service is planned and managed.

Testing is usually conducted in Service transition. Service operations takes care of problem, event and incident management. The process is continuing and after operations, design starts again. Continual process improvement covers all services (ITIL, 2007. p.3-7).

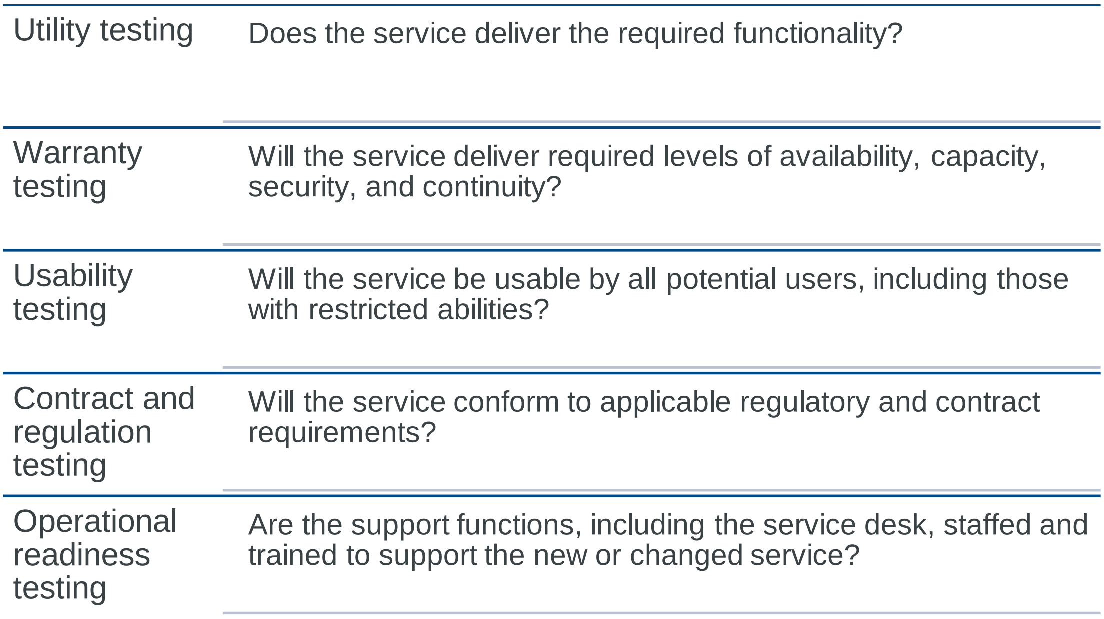
4.1.1 Service transition

Change management, Knowledge management, Release & Deployment management, Service Testing & Validation and Configuration management system are part of the Service transition. All of these supports the lifecycle process and creates value to the service. The goal is plan and manage service changes and deploy service release into production environment successfully. (ITIL, 2007. p.7)

Change management ensures that changes are roll-outed with minimum risk and without impacting to the service-levels and user experience. Changes are divided into standard changes, normal changes and emergency changes. Standard change is known beforehand and does not need additional permission. Normal changes have high risk and they go through change process, are reviewed by change advisory board (CAB) and gets authorization from change manager. Emergency changes requires actions immediately thus major incident behind it affect the users. Instead of CAB the permission comes from Emergency CAB (ECAB). (ITIL, 2007. p.42-61).

Set of authorized changes is a release which can be a “Big bang” or a phased roll-out. Big bang releases the service to user at once. Phased approach deploys the changes in several parts. Major releases contain new hardware or software and it is usually named v1.0 and v2.0. Minor releases improve the current functionalities and is v1.1 or v2.1. Emergency releases fixes errors apace. Those are named v1.1.1 and 2.1.1. **Release and deployment management** takes care of these and process around them, this includes also testing the release. (ITIL, 2007.p.84-86)

Testing the changes and new services makes a **Service validation and testing**’s assignment. Testing process includes planning the process, planning and designing tests, scheduling, preparing the test environment, performing the tests, evaluating exit criteria and report and cleaning up and closing tests. (ITIL, 2007. 9.115-116). The service design package (SDP) tells which tests to carry out. Types of tests include:



Utility testing	Does the service deliver the required functionality?
Warranty testing	Will the service deliver required levels of availability, capacity, security, and continuity?
Usability testing	Will the service be usable by all potential users, including those with restricted abilities?
Contract and regulation testing	Will the service conform to applicable regulatory and contract requirements?
Operational readiness testing	Are the support functions, including the service desk, staffed and trained to support the new or changed service?

Figure 9. The service design package test types (BMC, 2018)

As Figure 9 presents, Utility, warranty, usability, contract and regulation and operational readiness testing covers the SDP test types. Naturally, testing includes also traditional testing levels when required. (bmc, 2018)

Maintaining the service knowledge management system (SKMS) is **Knowledge management**'s main purpose. SKMS includes service portfolio, configuration management system (CMS), Supplier and contract management information system (SCMIS), Availability, capacity, and security management information systems (CMIS, AMIS, and ISMIS) and continual service improvement (CSI) register. Employees should have access to all information that their role enquires. This help team's decision making throughout the process if data is accurate, reliable and trustworthy. (ITIL, 2007. p. 145-146)

Service assets and configuration management's **(SACM)** responsibility is to ensure the assets required to deliver services are properly controlled and that correct and reliable information about those assets is available when needed. This optimizes organization's work and highlight assets that are vital to running of the customer's or organization's business. (ITIL, 2007. p. 65).

4.1.2 Roles and responsibilities

ITIL has roles for each function, for example, change management has a change manger, coordinator and initiator. A team does not need to have all ITIL roles. Change manager can be the only role for change management but also responsible of release and knowledge management.

Roles and responsibilities for each team member can be divided with RACI model to clear the roles and help resource allocation. RACI abbreviation comes from words "Responsible (R), Accountable (A), Consulted (C), Informed (I). The matrix is presented on Table 4 (Smartsheet, 2018).

Table 4. RACI model

	Team member 1	**Member 2**	**Member 3**
Task 1	R	A	C
Task 2	I	R/A	
Task 3	R/A	I	C

As seen on Table 4, on the left side is list of tasks, on the upper row roles and on the crossover is the participant type. Responsible person conducts the task. Accountable is the owner of the task who has the final authority/accountably. Responsible and accountable can be the same person when it is marked R/A. Consulted member shares knowledge and helps the accountable with decisions. There should not be too many people, therefore it can be time consuming and leads to poor quality. Informed people need to know the statuses of the task since it can affect their own work. However, it may be difficult to identify all who should be informed. Members can have multiplied roles in one task as mentioned before. A task does not need to have all roles but accountable is compulsory for all. (Smartsheet, 2018)

4.2 Testing

IT best practice suggests that software testing should be performed while developing the product. “Test levels” tell what kind of testing should execute in which phase. “The box approach” and “testing through lifecycle” describes best practices to do the testing. “When to stop testing” give list of good reasons to end the testing period.

4.2.1 Test levels

Test levels are stages of the software development. They are ensuring systematic testing which covers all aspects of the software. Unit testing is the first step to test the code. Two or more units together is a module which is tested with the component testing. The integration test makes sure that modules work together, and system test the compatibility of the application with the system. User acceptance testing (UAT) is carried out by customer to make sure that requirements are met. (ISTQB, 2018).

Unit testing is an on-going process during development. Developers test their own code and try to identify errors on the early stage. Code should also match to its design, requirements and functionalities. Changes are easier to make before a unit is connected to other units. Ensuring that units performance together, testers execute the component tests. (ISTQB, 2018).

Modules may work separately but not together, therefore **integration tests** need to be conducted. Testing reveal defects in the interfaces and in the interactions between integrated components or systems. (ISTQB, 2018). The big bang, top down and bottom up are three ways to perform integration tests, as shown in Figure 10 below.

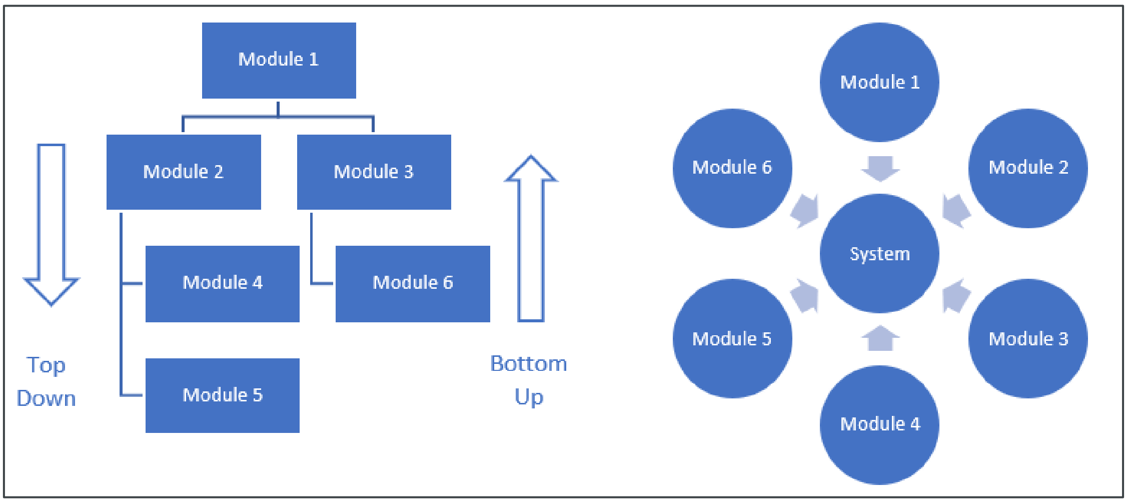


Figure 10. Integration test: top down, bottom up and the big bang

As seen in Figure 10, Top Down and Bottom Up tests have the same idea but they are conducted from different direction. Top Down testing starts from the top-level modules and sub-modules are added after the previous test is passed. The setting resembles the reality. Bottom Up testing starts from the sub-modules and higher-level modules come later. This allows developing at the same time. In both tests it is possible to use drivers/dummies to replace modules (Test institute, 2018). If all modules are ready and integrated, with the Big Bang testing it is possible to test the whole system. Although, tracking defects is harder than on Top Down and Bottom Up testing, thus everything is tested concurrently. (ISTQB, 2018)

System testing considers the whole integration in a one piece and investigates its functionality and performance. By executing different tests it is possible to find out the last defects and improve the software, before handing the product to the client. The most common test types are functionality, recoverability, interoperability, performance, scalability, reliability, regression, documentation, security and usability testing. These ensuring all scenarios are working as expected. (Software testing help, 2018)

User acceptance testing (UAT) is the last stage of testing, there the client approves that the software fulfils requirements and works in real working environment. The product is tested with actual employees, processes, computers and connections with other software. Test team can assist the client and create test cases to ensure the best outcome. Changes are possible to the software and after this testing level the production is ready to be released. (Hambling and van Goethem, 2013, p.15-18).

4.2.2 Box testing

Testing can be divided to Black and White box testing which describes the approach to the testing and test cases. Black box concentrates on finding circumstances where the software does not behave according to its specifications. Tester does not have access to the code therefor anyone can conduct the tests. White box testing is the opposite. It is based on the internal paths, structure and implementations. Grey box is hybrid of Black and White box testing. All approaches can be used on unit, integration and system testing. (Copeland, 2003. p. 8).

Black box testing starts with analyzing requirements and specifications, which make the basis of the testing. Then the tester chooses valid (positive test scenario) and invalid (negative) inputs for testing where the software process and defects them correctly. The next step is to determine the expected outputs for all inputs and conduct the tests. If actual and expected output does not match, there is a defect that should be fixed. After reparation there will be a new test. (Copeland, 2003. p. 19).

The disadvantage of the Black box testing is the quandary how much is tested since the tester does not see the code. Therefore, even several combinations of valid and invalid inputs with various data may not discover all defects. This drives to do efficient test cases to find the defects. (Glenford, 2011. p. 9-10).

Analyzing the software’s implementation is the first step of White box testing. After that the next step is identifying paths through the software and choosing the inputs for those paths. Expected result are determined before the tests are run. Afterwards the actual output and the expected output is compared to find defects. The number of logic paths can be too large for testing and still there can be paths that are not discovered in the identifying phase. Importantly, the tester needs the coding skills for executing White box test. (Copeland, 2003. p. 144-146).

In Grey box testing, the tester has some knowledge of the software’s structure and implementations. This helps creating comprehensive test cases by using the best sides of both approaches. (Homès, 2012, p. 144).

4.2.3 Testing throughout the life cycle

The application released is not the end of the software development. It is possible that users find faults or use the application differently than developers have thought. New features are needed and maybe the purpose or the appearance of the application change over the years. For example, Agile software development is based on several development sprints. All modifications lead to a new testing round.

Software maintenance is often fixing issues and doing changes to the software as a result of an external events. This development phase has two constrains: development and time. However, testing maintenance has timing (development and testing) and impact (ensure that other functionalities are not impacted) as a constraint. (Homès, 2012, p. 74).

After the default is spotted and corrected, the software confirmation and regression tests should be executed. Confirmation test repeats the situation or test case where the defect is noticed. The purpose is to verify that the feature works properly. Regression test ensures the functionality of the software. As seen in Figure 11 below, the components of the software are linked to each other. (Homès, 2012, p. 72-73)

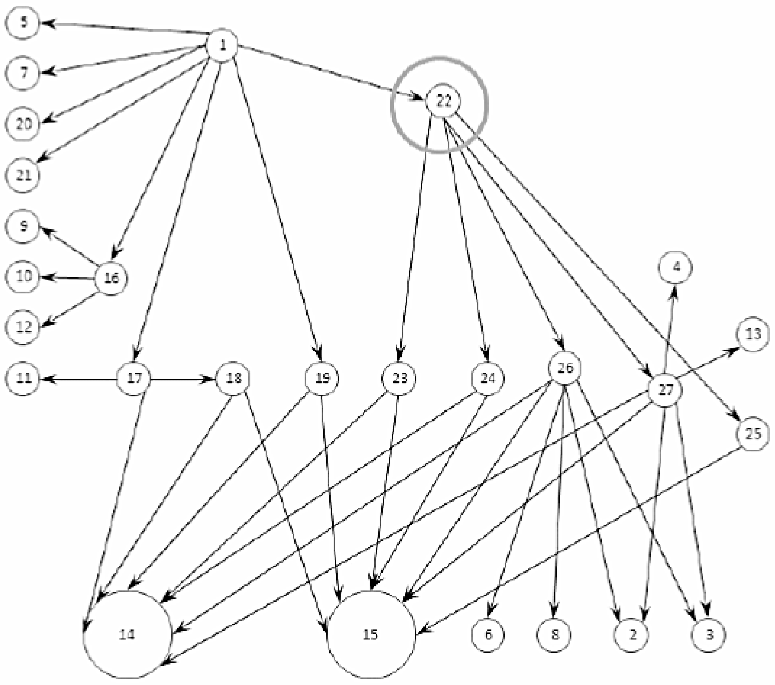


Figure 11. Impact analysis on software development (Homès, 2012, p. 73)

Figure 11 present how components could be arranged in the software. Changes to 22 may affect to 1, 23, 24, 25, 26 and 27. Besides testing these it would be reasonable to test components 14, 15, 6, 8, 2, 3, 4 and 13 since they could be indirectly impacted. (Homès, 2012, p. 72-73)

Agile software development has several quick iterations and it is open to the feedback and changes. The idea is that after every iteration the product is ready for the release. To ensure fast and high-quality deliveries, testing needs to be conducted in all iterations. IT best practice shows that bugs are easier to spot and fix in the early stages. Due to the amount of the testing, usually it is automated and test cases are created even before the code. (Measey and Radtac, 2015)

4.2.4 When to stop testing

Finding all defects from software is not financially reasonable. Testing is expensive and fixing faults can cause new ones. Copeland (2013) has listed five common reasons to stop testing:

- You have met previously defined coverage goals
- The defect discovery rate has dropped below a previously defined threshold
- The marginal cost of finding the “next” defect exceeds the expected loss from that defect
- The project team reaches consensus that it is appropriate to realise the product
- The boss says, “Ship it!” (Copeland, 2003, p. 237)

As a part of the test plan, testing limit can be set. Coverage goal defines how many percent of test case should be performed. The defect discovery rate determines the suitable number of faults in a specific period. For example, testing stops when there are less than 5 faults per week. Both criteria have a risk that simple test cases will lead to a full stop of the testing, although severe defect are not found. Also, a used time for testing will affect the number of found faults. The marginal cost of finding the “next” defect exceeds the expected loss from that defect. (Copeland, 2003, p.237-238)

Last two points do not have mathematical background, they are based on feeling. An experienced project team knows when most of the faults have been found and the product is ready for release. The boss can also end the testing process. Sometimes is important to get the software to market even though it has defects left. Although, effects on release and business need to be evaluated. (Copeland, 2013, p.239-240) As Hambling and van Goethem (2013. p. 162) summarize: “Combination of different reason is the best reason to stop testing and when it is difficult to justify continuing.”

4.3 Test management

Test manager is responsible for test management. It includes for example planning the process, test cases and managing the test management tool, reporting and decision of outsourcing the testing.

4.3.1 Planning

Test plan can cover a whole testing process (master test plan), or just one testing level, or type which test or project manager writes to clarify the testing. Usually, master test plan includes several levels or type specific test plans if those are made. The plan describes the scope, approach, resources, schedule and test activities. All features to be tested, environment, testing techniques, roles and testing tasks are listed to the plan, too. (ISTQB, 2018)

When writing a test plan, the workload must evaluate. According to Homès (2012), the process related, hardware, human and other delaying factors impact the workload. Process related factors include: constantly executing tests, changing management, process maturity, development and test processes, previous test phases, planned and actual levels of defects and corrections. Hardware factors are related to the software, such as tools, system tests, test environments and similarity with other projects. Human factory counts tester abilities and expectations, support from development teams and relationships between teams. Other delaying factors are, for example, complexity of the software, large number of stakeholders, too many new features and fragile test data. Also, the correct understanding of the estimation techniques and other aspects that may influence results. (Homés, 2012. p. 115-116)

The workload affects to the schedule, too. If test cases are divided evenly between the testing weeks, there is no proper time to do regression and confirmation tests (retest). Therefore, it is better to break testing period into parts as shown in Figure 12 below.

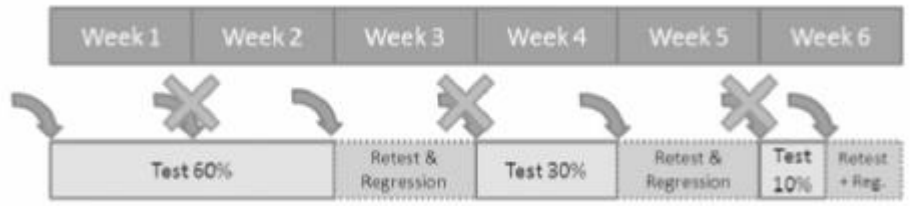


Figure 12. Non-uniform test distribution (Homès, 2012. p. 217)

Figure 12 shows that during first two weeks is reserved for 60 % of all test cases. These should include potentially critical and major cases. After first testing period and repairing, it saves a week for retest and regression test. The same process repeats during weeks four and five with 30 % of the test cases. The last week is saved for the last tests and the product should be ready for the delivery. With this technique, the testers have time to conduct all tests and report the defects without changes in the schedule or the scope. (Homés, 2012. p. 116-117).

4.3.2 Test cases

Crucial part of successful testing is the test cases. They can include a short sentence or comprehensive instructions. Test cases can be on an Excel file or on a test management tool. Each test case should have a reason for its existence. The test design process can be presented as a hierarchy as shown in Figure 13 below.

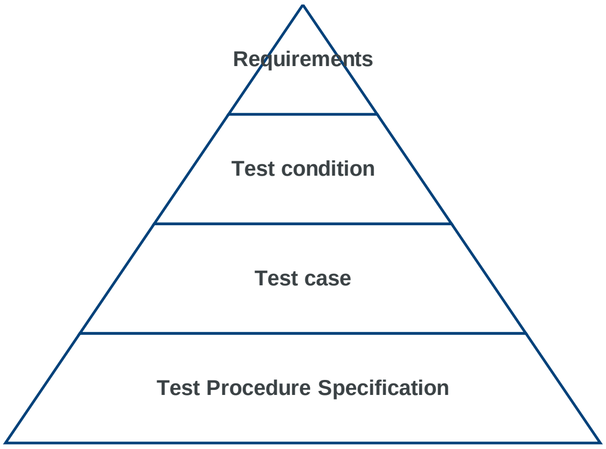


Figure 13. The test design hierarchy (Hambling and van Goethem, 2013)

As seen in Figure 13, the test design hierarchy includes four levels: requirements, test condition test case and test procedure specification. First step of the test design process is turning requirements into test conditions which describe what can be tested. One requirement can have several conditions therefor is substantive to use reference numbers. This enables traceability. (Hambling and van Goethem, 2013, p.123).

Test condition can be true or false after executing the test and to fulfil requirements all conditions should be true. A matrix is one way to verity that all requirements are covered. It contains a Design reference, reference number, requirement name and conditions. A test condition delineates one component of a feature for example “if a valid username is entered with correct password, the user is logged in”. Next conditions could be a situation where the user is not logged in since the password is incorrect or written in caps lock. Conditions may have some duplications to ensuring all situation are covered. (Hambling and van Goethem, 2013, p124-128)

Test cases have preconditions, inputs, expected outputs and post-condition. These give the tester an overall picture how to perform the test condition. Precondition and post-condition tell the start and end of the test and verify the correct expanse. If several test cases have the same precondition, such as “the user is logged in”, it can be an individual test which must carried out first. (Hambling and van Goethem, 2013, p.128-129)

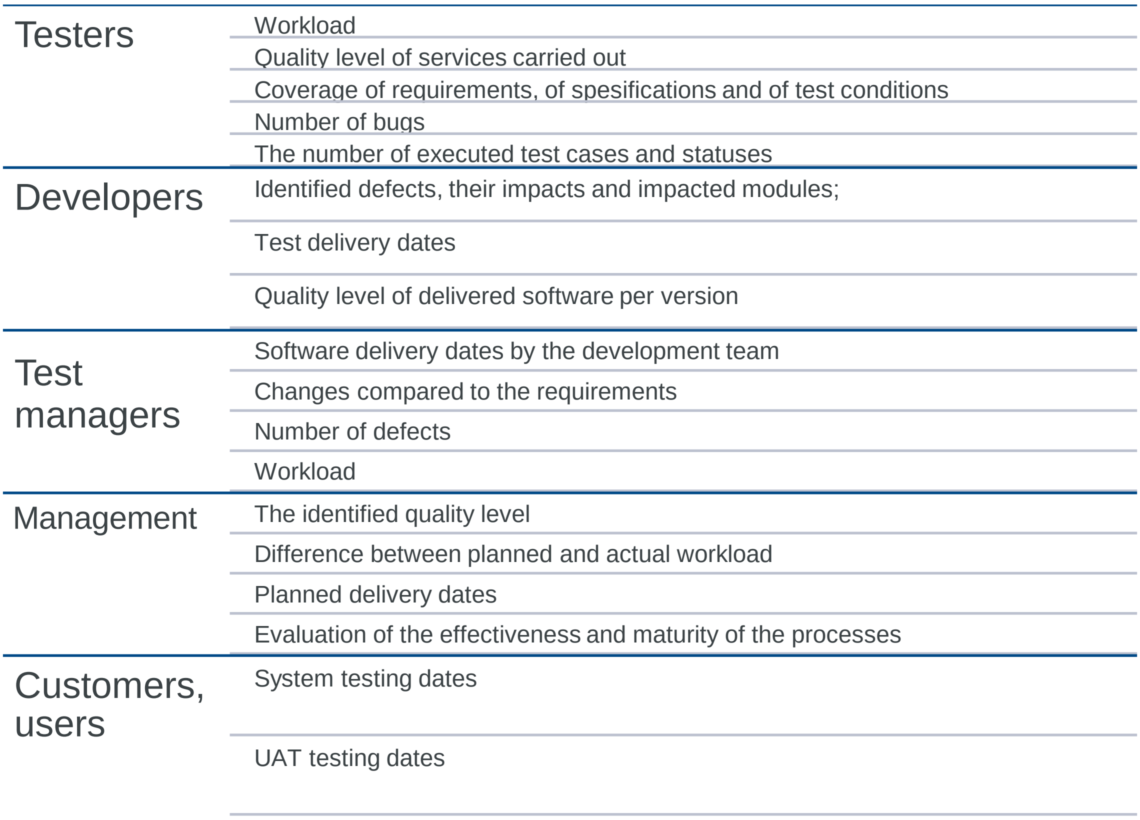
Defining the inputs for each test case will speed up the testing process. A tester does not need to guess, search information or ask credentials. The designed test data consider different roles, countries and other special cases. (Hambling and van Goethem, 2013, p.128-129). The input or output can be for example a specific view, a file or data from/to another source. The expected output and the actual output should match after test is executed, otherwise the test is failed. (Copeland, 2003).

A tester can conduct the test case without instruction if the system is familiar. Although, testing is faster and has better quality when test scripts are in use. Also, the tester does not need to know the software already. The test scripts should be simple and easy to follow. (Hambling and van Goethem, 2013, p.130). Part of the testing can be automated but not all since testing need creativity and curiosity to find all defects.

4.3.3 Test management tool and reporting

The decision between manual and automated testing defects on which test management tool to use. Also, there are all test levels tested with the tool and how the team's software are compatible to integration. (Mili and Tchier, 2015. p. 333) One option is to use MS Excel as a test management tool. This is a light option for small testing projects. Bigger projects, especially automated, need software for managing test cases, designate them to testers, reporting bugs and following the progression. Properly used test management tool helps the reporting.

Test reporting should target the stakeholders, thus the need of information is different. If the report is shared wildly, the information can be defined to the planned audience. Figure 14 shows the interests of different groups:



Testers	Workload
	Quality level of services carried out
	Coverage of requirements, of spesifications and of test conditions
	Number of bugs
	The number of executed test cases and statuses
Developers	Identified defects, their impacts and impacted modules;
	Test delivery dates
	Quality level of delivered software per version
Test managers	Software delivery dates by the development team
	Changes compared to the requirements
	Number of defects
	Workload
Management	The identified quality level
	Difference between planned and actual workload
	Planned delivery dates
	Evaluation of the effectiveness and maturity of the processes
Customers, users	System testing dates
	UAT testing dates

Figure 14. Stakeholders interests regarding test reporting (Homés, 2012. p. 242-243)

Figure 14 lists the interests of testers, developers, test managers, management and customers. Testers need to know how well they have stayed in the plan, and if the coverage is wide enough. Testers and developers are both interested of defects and the quality level of the software. Also, developers need to know the test delivery dates. Test managers use report to track the project and planning the future. They need information about schedule, workload, number of defects and changes compared to the requirements. Management has same interests as test managers but on a higher level. Customers and users are part of the UAT testing and sometimes also system testing, therefore they need to know specific dates for those. (Homés, 2012. p. 242-243)

4.3.4 Outsourcing the application development and testing

IT services or single application can be in-house or outsource to professional developers. According to Deloitte’s survey (2016), 59 % of companies say that cutting the costs is reason for outsourcing. Other common reason (57 %) is that it allows the company to focus their core business and employees to their actual job. Although, outsourcing needs more governance, clear contracts and responsibilities to meet the target levels. 64 % of replies are working to improve their Vendor Management Office (VMO) capabilities. (Deloitte, 2016).

Difference between the traditional IT outsourcing and the offshore outsourcing is in size. Outsourcing can be a whole IT department but offshore typically relates to a smaller area, for example, an application. Offshore usually indicates that work is conducted outside the borders. However, the theory can apply also for domestic outsourcing. Table 5 shows which operations are usually onshore and offshore when offshoring.

Table 5. Onshore vs Offshore for each IT stage (Gold, 2005. p. 9)

IT Life-cycle Stage	Primarily Onshore (in-house)	Primarily Offshore	Equally Onshore and Offshore
Business Partnership	X		
Overall IT strategy	X		
Application planning and design	X		
Business process planning and design	X		
Application analysis and design	X		
Application efficiency analysis/upgrades		X	
Application coding		X	
Application testing		X	
Application maintenance		X	
Application retirement planning			X

As seen on Table 5, the IT department stays in the company and it has authority to plan and create strategies, partnerships and processes. Also, application planning, analysis and design make part of the Onshore. Offshore takes care of the application development. It has efficiency analysis, upgrades, coding, testing and maintenance of the application. Application's retirement planning relates to for both parties. (Gold, 2005. p. 6-10).

The level of the offshoring is presented usually "80/20" or "70/30". The first number describes the percentage of offshore operations, and the last one of the onshore. Application maintenance is usually the first one to offshore. Then programming traditional languages (e.g. COBOL) and programming cutting-edge languages (Java). These are easy to offshore to Asia where expenses are lower. Relationship, program and project management and complex architecture is rarely offshore to abroad. They are kept in the company or onshore to local business. (Gold, 2005. p.8).

Testing can be also outsourced to improve the testing quality and get a new perspective. Payed testing team could reduce the workload from developers and do all testing that does not inquire changing the code. Other possibility is to hire a team to do only a specific test such as a security testing. (Gold, 2005. p. 6-10).

Vendor management office (VMO) creates strategic value to the company by optimizing vendor relationships. Legal, procurement, IT and business units work together to evaluating suppliers' goods and services. Benefit from vendor management is controlling costs, increase quality and lower the risk. Also, the company can centralize procurement and contract management, improve vendor relationships and regularize outsourcing.

4.4 Conceptual Framework

Based on the exploration of existing knowledge in Section 4 above, the following best practice of testing and test management help creating the Proposal. Conceptual Framework in Figure 15 presents the approach selected in this study to improve test management:

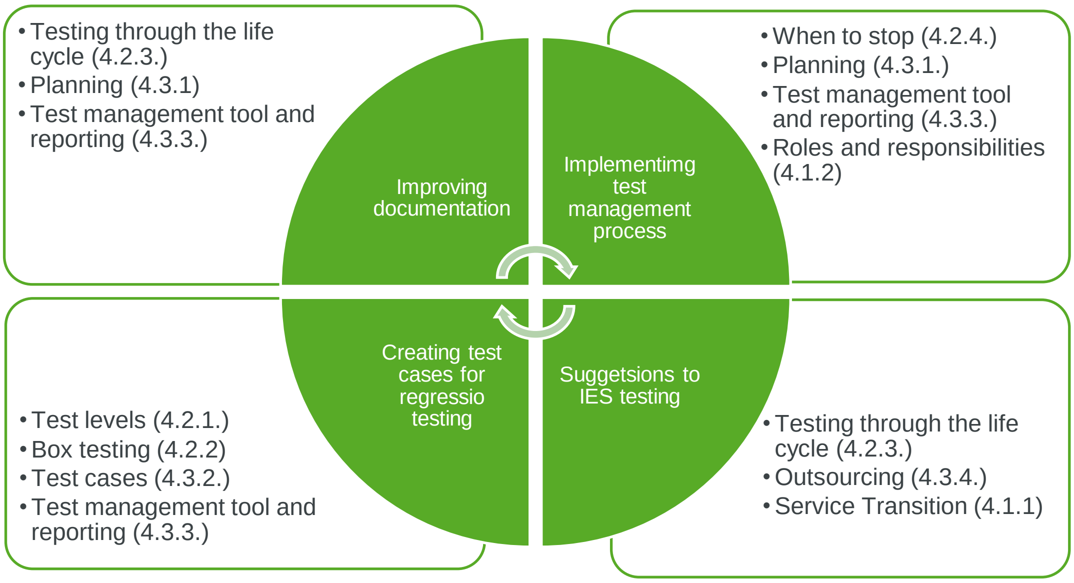

• Testing through the life cycle (4.2.3.)
• Planning (4.3.1)
• Test management tool and reporting (4.3.3.)
• When to stop (4.2.4.)
• Planning (4.3.1.)
• Test management tool and reporting (4.3.3.)
• Roles and responsibilities (4.1.2)
Improving documentation
Implementimg test management process
Creating test cases for regressio testing
Suggetsions to IES testing
• Test levels (4.2.1.)
• Box testing (4.2.2)
• Test cases (4.3.2.)
• Test management tool and reporting (4.3.3.)
• Testing through the life cycle (4.2.3.)
• Outsourcing (4.3.4.)
• Service Transition (4.1.1)

Figure 15. Conceptual framework of the Thesis

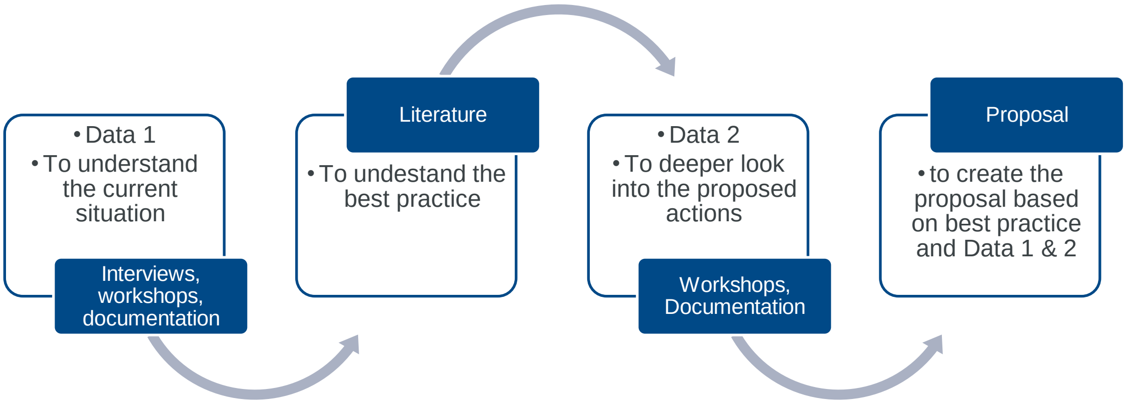
As seen in Figure 15, proposal suggests improvements in four areas: improving documentation, implementing test management process, creating test cases for regression testing and suggestions to IES testing. Each area is linked to the best practices and relevant theory. Next section presents the Proposal and how it was conducted.

5 Building the Proposal

This section builds the proposal of implementing test management process and improving the quality of testing. The ground of the proposal is the current state analysis and the best practices from literature view.

5.1 Overview of the Proposal Building Stage

The goal of this section is pooling together the key findings from Data 1, Data 2 and suggestions from existing best practice summarized in the conceptual framework. They underlay the proposal building and lead to formulating the actual proposal which contains three elements: (a) implementing test management process, (b) improving documentation and (c) creating test cases. Figure 16 below describes how the proposal is built and the logic guiding the proposal building.



• Data 1
• To understand the current situation
Interviews, workshops, documentation
Literature
• To undestand the best practice
• Data 2
• To deeper look into the proposed actions
Workshops, Documentation
Proposal
• to create the proposal based on best practice and Data 1 & 2

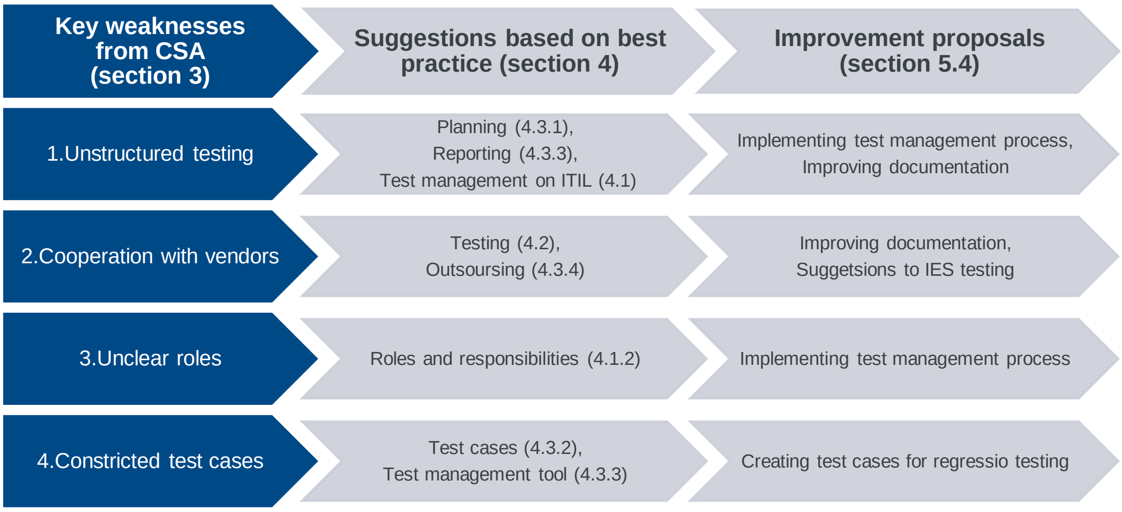
Figure 16. The logic of building the proposal

As seen in Figure 16, the proposal is built in four stages. Data 1 was the basis of the current state analysis and topics of the literature study. Workshops and interviews gave the overview of the current test practices and problem areas. Getting to know the case company’s documentation and applications helped understand the working environment. After identifying the problem areas, the search focused on best practice on testing and test management, so that to find the right directions for improvement and helped to prepare for the proposal building.

Based on these suggestions, Data 2 collection was conducted that included more specific look into the case company’s material and studied the testing practices of other teams. Findings from these workshops and documentation helped to formulate the proposal how to modify the current state of testing in the case team.

5.2 Key Findings from Data 1 from the Current State

This section and Figure 17Figure 17 summarize the key findings and related theory which made the basis for the improvement proposal.



Key weaknesses from CSA (section 3)	Suggestions based on best practice (section 4)	Improvement proposals (section 5.4)
1.Unstructured testing	Planning (4.3.1), Reporting (4.3.3), Test management on ITIL (4.1)	Implementing test management process, Improving documentation
2.Cooperation with vendors	Testing (4.2), Outsoursing (4.3.4)	Improving documentation, Suggetsions to IES testing
3.Unclear roles	Roles and responsibilities (4.1.2)	Implementing test management process
4.Constricted test cases	Test cases (4.3.2), Test management tool (4.3.3)	Creating test cases for regressio testing

Figure 17. Key findings from the current state vs. theory vs. proposed improvements

First, the current state analysis (in Section 3) focused to understand how the test management is conducted at the IOS team. The most important findings from workshops are incomplete documentation, lack of test cases and unstructured testing process. Data was gathered from interviews, documentation, applications and workshops. These workshops, interviews and documentation gained valuable knowledge for identifying key weaknesses in the current testing practices.

Unstructured testing, cooperation with vendors, unclear roles and *constricted test cases* were identified as main weaknesses of the current state. It was found that:

Documentation is kept in vendors’ project portals and SharePoint. Vendors do not provide test reports after testing. Team members are unsure how comprehensive the testing has been and are all areas covered. Afterwards it is difficult to investigate old testing results. Team do not use specific test management tool, but they report results to vendors' project portals.

Also, the case team does not have proper test cases for regression testing. Testers know how to use the application, and which features cause incidents. They test those, and vendors also provide list of new features to test. Testing is impossible to people who do not understand the software.

Finally, the test management process is not planned and pictured before. Therefore, roles and timelines are unclear which cause lack of testing time before releases. IT operational handbook shows the case company's testing process, but it is not directly transportable to the case team.

In the next step, the theory was explored to address these identified challenges. The relevant improvements proposed based on best practice and literature are discussed below.

5.3 Suggestions from Existing Knowledge and Literature (Conceptual Framework)

Each weakness that was identified in the current state analysis was analyzed separately. Suggestion were searched for from best practice and literature (Section 4) to improve each of the key weaknesses. This search also gave valuable knowledge about basic concepts of testing.

The first key finding from the current state analysis was unstructured testing. Sub-sections planning (4.3.1), reporting (4.3.3) and test management on ITIL (4.1) gave knowledge to tackle this weakness. The second finding was cooperation with vendor. The best practices came from testing (4.2) and outsourcing (4.3.4). Unclear roles was the third point, and in literature its counterpart was roles and responsibilities (4.1.2). The final weakness was constricted test cases. Sub-sections test cases (4.3.2) and test management tool (4.3.3) gave expertise to create test cases.

Test levels and box approach help to understand what kind of testing the case team does and why. Test management section presents the best practices for example creating test cases and creating documentation. Theory of outsourcing the application development and testing concerns all application, especially IES. The IT operating model is based on ITIL, therefore, it is necessary to understand the basic concepts.

In the next step, based on the suggestions from best practice and literature, the proposal was formulated how to address the identified challenges. Data 2 for the proposal building was collected from the interviews, workshops and documentation in the company.

The improvement proposal is described below.

5.4 Proposal

The proposal for improving test management in the Installation Operation Solutions includes four elements in Figure 18: (A) structuring test management process, (B) improving documentation, (C) creating test cases, and (D) suggestions for IES testing.

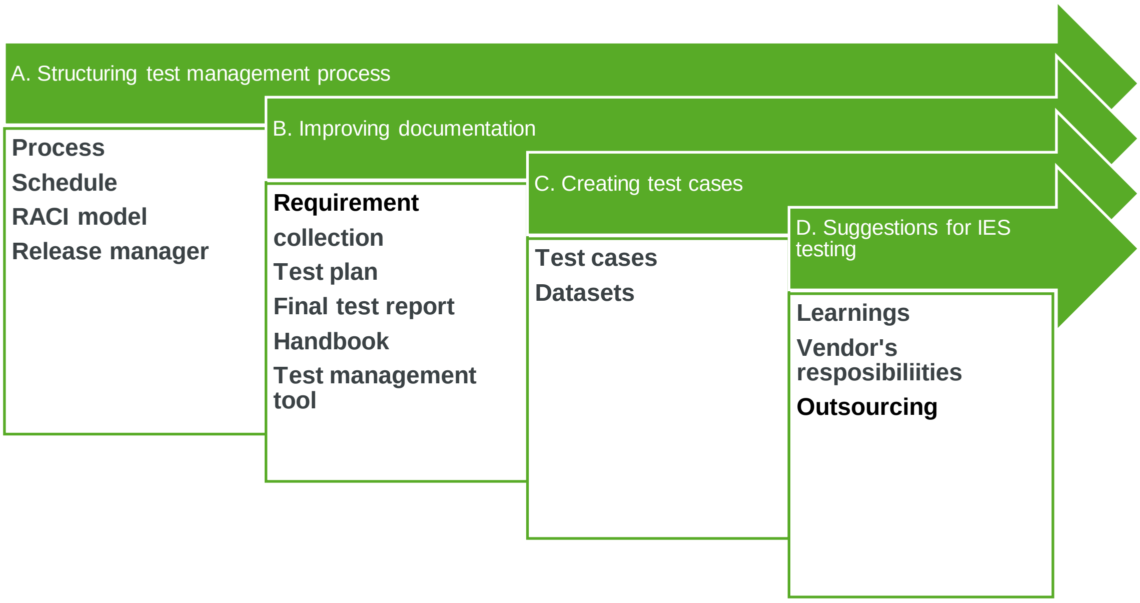


Figure 18. Proposal for improving test management process

Figure 18 presents the proposal and order for deploying elements into the IOS team. (A) Test management process tells how to implement the IT operational handbook's model and defines the roles. This is the foundation for other elements and request from the IOS team. (B) Improving documentation presents which documents should be created during testing and where to store those. (C) Creating test cases will improve the quality of testing. Finally, (D) suggestions for IES testing and learnings from the team are presented in the last part of the proposal. The topics came from the current state analysis and workshops with the team. Process, schedule, roles, test management tool and vendor's responsibilities were the suggested topics from the IOS team.

The key strengths from the current state analysis were also considered. *SharePoint is used, IT Operating model is launched at the case company* and *IRMA and IBR have test cases* are concrete strengths that help deploying the proposal. *Employees are ready to improve* and *employees are used to do testing* help the IOS team along the transition.

5.4.1 Structuring test management process

First*, implementing the case company's IT Operating model* and its test management process will structure team's work and afterwards deploy the other parts of the proposal. The IT Operating model states how the test management process should be conducted for the service pack releases (Appendix 2). The IOS team does not follow the process currently. Though the original process is too specific and arduous for the IOS team, the proposal presents the streamlined process. It has the same main phases and scope as the original. Figure 19 shows the modified process and Appendix 3 the schedule.

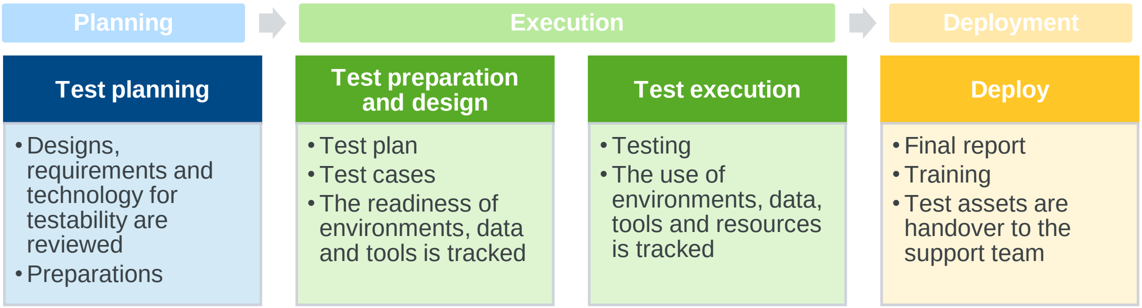

Planning
Execution
Deployment
Test planning
• Designs, requirements and technology for testability are reviewed
• Preparations
Test preparation and design
• Test plan
• Test cases
• The readiness of environments, data and tools is tracked
Test execution
• Testing
• The use of environments, data, tools and resources is tracked
Deploy
• Final report
• Training
• Test assets are handover to the support team

Figure 19. Test management process of the service pack release

As seen in Figure 19, the main process has Planning (blue), Execution (green) and Deployment (yellow). Colors are borrowed from IT operating model and used also in the thesis.

As seen from Figure 19, the test management process is divided into three parts and four phases. The first part, *planning* is for reviewing designs, requirements and technology for testability. This could happen together with the vendor. Also release and test plan is created, and resources are committed to the testing period.

Second part is called *execution*, it has two phases: *test preparation & design* and *test execution*. In the first phase, test plan is updated, and the readiness of environments, data and tools is tracked. Test cases are prepared or reviewed if the vendor has created those for the testing. The second phase, test execution, is the actual testing. There the focus is testing the design and implementation of the solution to the specification. All test levels are tested but the vendor should also do part of it. Key-users do the UAT testing before informing end-users about the release. The use of environments, data, tools and resources is tracked during the testing.

Third part is *deploying.* Right after the release is trainings or the info-call session to all the key-user to introduce the latest version. The final report is reviewed and signed off. It contains all planned test levels, reports from testing teams and the lessons learned document. Test assets such as the test plan and test cases are archived and handed over to the support team. The satisfaction survey is sent once a year to all the users.

These three parts above have described the proposal for the test management process for the service pack release.

Next, *defining roles and responsibilities for the phases* should be the second part of the process implementation. RACI model (Table 6) specifies responsible (R), accountable (A), consulted (C) and informed (I) person for each task.

Table 6. RACI Model for test management

RACI		Solution Owner	Solution Design Owner	Service Manager	Configuration Owner	Release manager	Rollout Coordinators	Tester	CAB Meeting
Planning	Requirements	C	R/A	C	C				
	Release scope	I	R/A	(R)	(R)				
Execution	Test & release plan	I	(R)		(R)	R/A		I	
	Test cases		I			R/A		C	
	Release validated		X			R/A	X		X
	Testing	I	C		C	A		R	
Deployment	Release deployed (training)		C	C		R/A	R		
	Feedback (once a year)	I	A	C		R	R		
	Final test report	I	I	I		R/A	I	I	
	Release review	X	X	X	X	R/A			X

As seen in Table 6, the case team has Solution Owner, Solution Design Owner, Service Manager, Configuration Owner, Release Manager, Rollout Coordinators and Tester. CAB-meeting (Change Advisory Board) gathers twice per testing period. All roles, except for the release manager and tester, exist already but their responsibilities in test and release management are not defined.

First, *Planning* stage includes requirement collection and defining scope of the release. Solution design owner is accountable and responsible for both. Service manager and configuration owner helps with defining the scope, and they are also consulted with solution owner about the requirements.

Second, *Execution* is mainly the Release manager’s area. She/he creates the test and release plan together with Solution design owner and Configuration owner. Tester is consulted with test cases but otherwise it is for the Release manager’s job to prepare or review them. Release validation happens in CAB-meeting. Tester does the testing, but Release manager is accountable. Solution design owner and Configuration owner can be consulted.

Third, *Deployment* takes place after the service pack is released. Training after release and collecting feedback once a year is the responsibility of the Rollout coordinators and/or Release manager. Release manager is accountable of the training and Solution design owner of the feedback. Release manager also writes the final test report and gathers the CAB meeting for reviewing the release.

Finally, *the case team should have Release manager and a specific Tester*. According to the workshop conversations, Release manager would be the most efficient role to manage release and implement change management. Tester’s role could be assigned to a team member or members with every testing. Most likely, Rollout coordinators do the testing at beginning. New customer service team will start at India during Autumn/Winter 2018. After that, it may be possible to reorganize the current roles and job descriptions to establish a Release manager role to the team. Customer service could also participate in testing.

Thus, the proposal for *Structuring the test management process* proposes three improvements: first, *implementing the case company’s IT Operating model*, second, *defining the roles and responsibilities for the phases* and, third, *the case team should have a Release manager and a specific Tester*. Next sub-section gives suggestions for improving the documentation.

5.4.2 Documentation

First, *ingenious test management* requires several documents. It would be practical to think of the structure again for SharePoint and consider what other tools could be used during the testing process. Developing starts with collecting needs and transforming those to requirements. The first step of testing process is creating a test plan. It is followed by creating test cases and instructions to the testers. During testing, all steps, issues and results should be documented and based on these documents, a final report should be created afterwards. After the release, the handbook should be updated, and training materials created for the users. The last step is collecting feedback form the users.

Second, *requirement collection* will move to Remedyforce, where the key-users can suggest changes to applications. This is a request from the case company. For tracking requirements, it would be useful to add an identify number for each requirement. This would also ease creating the test cases and updating those. With every release, the requirement list could be published to SharePoint for the users.

Third, *the IOS team can combine test and release plan into one document* at the beginning of implementing the new process. Currently, their test plan is only the vendor's plan and documentation. The propose is that the test plan part should be a master test plan, and it should focus more on those test levels which the team will conduct. It describes the scope, approach, resources, schedule and test activities. All the features that need to be tested, the environment, testing techniques, roles and testing tasks are also need to be listed in the plan. In the plan, the vendor's test plan should be an appendix if it is provided to the team. The release and test plan should be placed to SharePoint where the whole team can have access to it.

Fourth, *in-house test cases should be storaged into the test management tool* that the thesis worker has chosen and deployed for the team. However, since vendors do not have access there therefore, bugs must be reported to the vendors' project portal. This will cause extra work but managing test cases in an application will be easier than in the Excel sheet. Also, it will become simpler to assign the test cases to the testers and tracking the project.

Fifth, *release manager writes the final test report* after the testing period. It will present how the testing was carried out and report if it followed the test plan. Information comes from the testers, vendors and test management tool. It is also beneficial to add the learnings chapter to the final report to avoid similar mistakes later.

Sixth, *rollout coordinators update the handbook and sometimes give training sessions to the key-users after the release*. Trainings and releases is informed to the users by email and all material is put into SharePoint. This happens already after the release. The case company prefers that all instructions are stored in Remedyforce, thus all employees have access to these documents. The most convenient way would be to create a knowledge article template in RemedyForce and add a link to a SharePoint document. This way team will need to update the documents only in one place.

Finally, *the customer survey should send to the users once a year*. The team felt that sending customer surveys to the users is arduous. Thus the key-users need to translate all questions and answers to the local language and back. Therefore, it is not reasonable to send a customer survey after every release. The case company has a license for one survey tools which gives visual reports from results. The collected information thus becomes visible and valuable for improving the solution, trainings and the process.

Thus, the proposal of *improving documentation* includes the improvements related to the use of tools, document and planning the requirement, testing and release processes. Next sub-section concentrates on the proposal for the test cases in regression testing.

5.4.3 Test cases for regression testing

First, *the IOS team should have test cases for regression testing*. Currently, vendors provide test cases or test objects for system testing but not for regression testing. Regression testing ensures that all features work after the development, which makes it critical for the user experience and avoiding incidents. Also, with written test steps it is possible that the new customer service team in India could do part of the testing. Current test cases are too constricted for them. This would save time from the current team members for their daily work.

Second, *determining datasets beforehand for IMT testing* will make the testing more effective and comprehensive. IMT has different set-up for each country and language. Therefore, it is not realistic for the case team to test all features with all countries. IBR is in the pilot stage and presently has only two user countries. It is simple to do testing with both of these countries but IBR also needs a dataset table on future. This should be done as soon as the team has gained knowledge about the behavior of the application with different country setups and identified vulnerable features. IRMA testing is not as country sensitive as others.

Finally, *updating test cases should be assigned to Release manager*. Before every testing period, he/she must go through all test cases and check whether the content is still correct, and whether new cases should be created. Otherwise test cases are useless. These actions will enquire Release manager’s commitment and time. Numbering the requirements and connecting them to the test cases would also expedite this process.

Thus, the proposal for *creating test cases for regression testing* includes two improvements: first, *the IOS team should have test cases for regression testing, determining datasets beforehand for IMT testing* and, second, *updating test cases should be assigned to Release manager*. Next sub-section gathers the best practices and learnings to ease the IES testing.

5.4.4 Suggestions for IES testing

Installation execution system (IES) development project is currently in its beginning stage, therefore, it would be good to consider testing practices in advance together with the team and vendor. Also, best practice of the test management as well as learnings from the case team have proven that communication with the vendor is critical. Therefore, the following improves are proposed below.

First, *agreeing the best practices for the testing period* is necessary for everyone. Vendor and the case team can propose how they want to do the testing. Vendor probably has the company-level instructions for writing test scripts, reporting bugs and managing the project. The case team should express their needs about the delivery (test report, handbook, screenshots, etc), as well as staying on schedule and coverage of testing. Training session for the vendor's project portal and testing practices would also increase effectiveness of the IOS team's testing.

Second, *outsourcing part of the testing* could ease some of the team's workload. The current IOS team does not have resources for testing four different applications which are released twice a year. Therefore, it would be convenient to increase the vendors' responsibilities of testing. One team in the Delivery Solutions has outsourced all system and regression testing to the developer (meeting 6) and this team does only UAT testing together with business. This brought up some additional expenses, but Solution owner was satisfied with this arrangement. At least, the case team could use more outsourced testing possibilities to relieve some of the workload.

Third, *following the IT operating model, its project section will help in IES developing*. Numbering requirements and creating all needed documents will also clarify the testing process itself. The team has had time to assimilate best practice of the test management and content of the Proposal before IES comes to the testing phase. Therefore, testing will be more professional than before.

Thus, the proposal for *IES testing* includes three suggestions: first, *agreeing on best practice for the testing period*, second, *outsourcing part of the testing* and, third, *following IT operating model's project section will help on IES developing*. The next sub-section validated this initial proposal with the team, and presents the final proposal after incorporating the team's final suggestions.

6 Validation of the Proposal

This section describes the results of validation of the Proposal developed in Section 5. First, it gives an overview of validation stage. Then, it describes the feedback and evaluation of the Proposal. Based on the feedback the Final Proposal is crated on chapter 0.

6.1 Overview of the Validation Stage

The validation directs the proposal to its final version with small adjustments and it gave opportunity to the case team to influence the final proposal.

Data 3 for the validation has been gathered from two solution design owners, service manager, platform & configuration owner and solution owner. They are responsible for implementing the IT operating model to the case team. Communication with the customer ensured the reasonable and useful outcome also from the customer. Figure 20 shows the steps in the validation process.

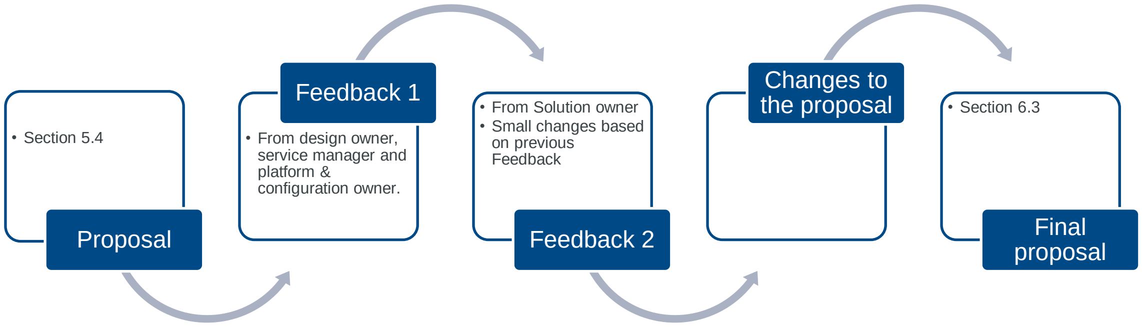

• Section 5.4
Proposal
Feedback 1
• From design owner, service manager and platform & configuration owner.
• From Solution owner
• Small changes based on previous Feedback
Feedback 2
Changes to the proposal
• Section 6.3
Final proposal

Figure 20. Validation process

As seen in Figure 20, the first part of the validation was a meeting with design owner, service manager and platform & configuration owner. There, the proposal was introduced and discussed.

The second part of validation was conducted a couple of months after the first validation and end of the project. This way the solution owner had also opportunity to give his opinion. Based on feedback the changes were made to the proposal and this way final proposal was created.

6.2 Results of Validation (Data 3)

The following points were stressed in the validation.

The proposal includes four areas: test management and roles, documentation, test cases for regression testing and IES testing. Test management process is lined with the IT operating model and possible to take into practice on the future testing periods. As noted in the proposal, the team needs to establish the role of release manager to follow the process. The team will reshape the team structure and changes the roles of rollout co-ordinator into Application specialist. The role of Indian customer service team is uncertain at this point.

Documentation needs effort from the team. Two reports with every testing period, updating requirements and test cases into SharePoint and collecting feedback from the users could be too much. This have been noted in the final proposal. Test cases was the biggest concern for the team. They need frequent updates and negotiations about the test management tool licenses is paused.

Suggested practices for IES testing were approved. The team was agreed that the role of the vendor needs to define clearly for testing. Following the IT operational model in first steps of the project has been successful therefore the model will be used on the project and later, on the Service area.

Also, one of the most important finding was that solutions need to be tested with different dataset to ensure functionality in all countries. With this proposal it could be possible to increase testing quality and save time.

6.3 Final Proposal

The Final proposal is based on further developments to the initial proposal (in Section 5.4) and the feedback from the team (in Section 6.2). It clears the documentation and present the status of actions as shown in Table 7 below.

Table 7. Done, ongoing and proposed actions with the responsible person(s)

Action	Responsible	Status
Selecting the test management tool	Thesis worker	DONE
Instruction’s for the test management tool	Thesis worker	DONE
Test cases	Thesis worker	DONE
Datasets for IMT	Service manager	ONGOING
Documentation templates	Service manager	ONGOING
Process implementation	The case team	ONGOING
Release management terminology	The case team	ONGOING
Schedule	The case team	ONGOING
Role of the Release manager	Solution owner	PROPOSED
SharePoint folder	The case team	PROPOSED
Negotiations with IES vendor	Solution design owner (IES)	PROPOSED

Table 7 shows the status and responsible person for each action. Responsibilities are agreed on the feedback meeting. Test management tool was selected, and training demo with instruction PowerPoint was given during the project. The test management tool is already used in the case company, therefor the case team hope to get free or cheaper licenses by using the same tool. Number of licenses was not cleared when the thesis project ended. However, test cases are already made for the tool for IMT (41 cases), IRMA (33 cases) and IBR (34 cases) solutions.

Based on the ticket analysis from the current state analysis and the team members’ insights, the outcome of the Workshop 14 (appendix 1) was datasets for regression testing (appendix 4). It determines the minimum testing for primary features and helps selecting users for the test cases. Key-users should do a UAT testing before informing the release to their end-users. This way all countries are still covered during testing. Next step for the team is to use agreed datasets and think, if the scope enough.

To ease the documentation, the team can use the case company's test plan and test report templates. Service manager will modify those for the team's needs. Process implementation is ongoing thus it is connected to release, change and requirement management which are not carried out. Next step for release management could be standardize the terminology and numbering for service packs. ITIL suggest using v1.1.1 for numbering (section 4.1.1).

The schedule is already on the SharePoint calendar for the next five years. Although, the dates are not exact and the release manager should agree those with vendor and the team. Also, other IT Operating model processes affect the dates. Otherwise, SharePoint is not shaped to the project. It was agreed that the team will do it later and think which documents they want there.

The key factor for the successful test management implementation is the role of the release manager. This did not happen during the project thus solution owner was not around to make the decision. Also, the actual role and benefit of new customer service team is not identified. Negotiations with vendors was not part of the project.

RACI model (appendix 5) is modified at request. Rollout coordinator and tester are combined into Application specialist. IMT and soon IRMA are roll-outed to all countries and therefor they do not have rollout coordinators anymore. Application specialist will have partly same job description as a rollout coordinator.

7 Summary and Conclusions

This section presents the executive summary with the project overview and its main conclusions. In addition, this section includes key findings, next steps and evaluation of the project.

7.1 Executive Summary

The objective of the thesis was to harmonize the testing processes together with the case team, vendors and the company.

The project was carried out in five steps. The project started by defining the objective, outcome and business challenge based on the discussion with the team. The next step was to understand the current state. The information was gathered from workshops with the team and in-house documentation. Learning to use all team's solutions and vendors' project portals gained also valuable knowledge for the current state analysis. Based on strengths and weaknesses on the current state analysis, the best practice and literature study was conducted. The best practice on test management helped to create the proposal. Also, meetings with test managers and solution owners from other teams were beneficial. The last step was gathering feedback from the team members and creating the final proposal based on the team's insight.

The key findings of the current state analysis can be divided into the strengths and weaknesses, as seen in Table 8 below.

Table 8. Strengths and weaknesses from the current state analysis

Strengths	**Weaknesses**
SharePoint is used	Cooperation with vendors
Employees are ready to improve	Unstructured testing
IT operating model is launched at the company	IMT testing process is not thought
Employees are used to do testing	No clear roles
IRMA and IBR have test cases	Constricted test cases

Table 8 shows that the biggest strength was using SharePoint by the team. Also, the team was willing to improve and knew how to test. Also, the case company had launched the IT operating model and vendor provides test cases to IRMA and IBR system testing.

The main weakness was that the testing process has not been defined. Unstructured testing was time consuming and roles were not clear, therefore, the case team wanted to improve the testing management. Therefore, cooperation with vendors, undefined the roles and constricted test cases were the problems.

The proposal tackled these weaknesses in all four problem areas. It included: (A) process and roles, (B) documentation, (C) test cases, and (D) suggestions for IES testing.

First, the proposed process is based on IT operating model's process, but it has been modified to respond to the needs of the case team. Testing schedule and RACI model of the roles make the core of this process. The role of the release manager was also identified as an essential part of the proposal.

Second, documenting the process is for helping the team. Test plan describes for example the schedule, roles and test cases which will clearer the testing process. Final test report tells what went well and what not. This is valuable information for the future development and testing. Also, using SharePoint will increase transparency within the team.

Third, test cases are meant for the regression testing to ensure that all functionalities work after the development. With these, testing is faster, more consistent and the tester does not need to know the application well. When using the test management tool, test cases can be assigned to specific tester and the progress can be tracked. These reports can be added to the final report.

Finally, IES testing is grounded in best practice for organizing the testing in the case team. The development project for IES has not started yet, therefore the team can negotiate with vendor how the testing should be conducted. Other advantage is following IT operating model processes is possible from the beginning.

The Proposal should increase the quality of testing and reduce the incidents after releases. Keeping the solutions usable always is important for the company's business and customer satisfaction. Also, since the case company had launched the IT operating model where testing management process is pictured, and all teams are expected to implement the operating model, this proposal helps the case team to implement the IT operating model in their area.

7.2 Next Steps and Tips for Implementation of the Proposal

Firstly, the proposal and test management are connected to release, change and requirement management, therefore it would be beneficial to think those processes thoroughly soon. This also includes the role of the release manager which is essential part of the proposal.

Secondly, more licenses are needed for test management tool to use it effectively. Therefore, negotiations with the contact person should continue.

Thirdly, the team should conduct the whole test management within the next testing period. This way they will know what to improve and keep or is something missing. It will take couple of rounds before the new process starts to feel natural and simple to conduct.

7.3 Thesis Evaluation: Objective vs. Results

The objective was to harmonize test management between the team, vendors and the company. The proposal gave the tools for the team to manage their testing process and company’s IT operating model is part of the proposal. Communication with vendors could have been a bigger part on the project to ensure harmonization. Partly summer holidays and changes in development team affected that interviews with vendors did not happen. Deeper look into vendors’ action and how team works with them could also have gained more valuable information of the current state. Also, it would have been reasonable to have one-to-one interviews with team members beside workshops.

From the point of view of the author, the proposal is logically built and ready to use in practice. Especially schedule and RACI model should make test management more concrete in effortless way. There are together over one hundred test cases in the test management tool and instructions for it. Hence using the tool should make it easier for the team.

As the final words from the author, test management is part of the bigger picture and a wonderful way to start organizing the team’s processes. Implementing test management will give structure to the testing periods and moreover will help communicating with vendors during the testing. Both will increase the team members’ satisfaction.

References

bmc. 2016. ITIL® Service Transition. Available from: https://www.bmc.com/guides/itil-service-transition.html (Accessed 18 October 2018)

Copeland, L. (2003). *Practitioner's Guide to Software Test Design.* Available from: https://ebookcentral.proquest.com/lib/metropolia-ebooks/detail.action?docID=227688#

Deloitte (2016). *Deloitte’s 2016 Global Outsourcing Survey.* Available from: https://www2.deloitte.com/content/dam/Deloitte/nl/Documents/operations/deloitte-nl-s&o-global-outsourcing-survey.pdf (Accessed 25 September 2018).

Gold, T. (2005). *Outsourcing Software Development Offshore: Making It Work.* New York: Auerbach.

Hamling, B. and van Goethem, P. (2013). *User Acceptance Testing: A Step-By-step Guide.* Swindon: BCS Learning and Development Ltd. Available from: https://ebookcentral.proquest.com/lib/metropolia-ebooks/detail.action?docID=1192409#

Homès, B. (2011). *Fundamentals of Software Testing.* New Jersey: John Wiley & Sons, Incorporated. Available from: https://ebookcentral.proquest.com/lib/metropolia-ebooks/detail.action?docID=1120766#

ISTQB. (2016). International Software Testing Qualifications Board web site. Test management. Available from: https://www.istqb.org/certification-path-root/expert-level/test-management.html (Accessed 29 August 2018).

ISTQB. (2016). ISTQB Foundation Level and Agile Tester Certification guide web site. What are Software Testing Levels? Available from: http://tryqa.com/what-are-software-testing-levels/ (Accessed 29 August 2018).

Mili, A. and Tchier. F. (2015). *Software Testing: Concepts and Operations.* New Jersey: John Wiley & Sons, Incorporated. Available from: https://ebookcentral.proquest.com/lib/metropolia-ebooks/detail.action?docID=4040909#

Myers, G. and Co. (2011). *The Art of Software Testing.* New Jersey: John Wiley & Sons, Incorporated. Available from: https://ebookcentral.proquest.com/lib/metropolia-ebooks/detail.action?docID=697721#

Smartsheet. (2018). A Comprehensive Project Management Guide for Everything RACI. Available from: https://www.smartsheet.com/comprehensive-project-management-guide-everything-raci (Accessed 17 October 2018).

Software testing help. (2018). What is System Testing – A Ultimate Beginner’s Guide. Available from: https://www.softwaretestinghelp.com/system-testing/ (Accessed 29 August 2018).

Watts, S. (2018). What is an IT Vendor Management Office (VMO)? Available from: https://www.bmc.com/blogs/vmo-vendor-management-office/ (Accessed 18 October 2018).